

<http://dx.doi.org/10.17703/JCCT.2023.9.3.803>

JCCT 2023-5-95

센서·OPC-UA 시뮬레이션을 통한 엣지 기반 경량화 플랫폼 스토리지 엔진 평가

Evaluation of Storage Engine on Edge-Based Lightweight Platform using Sensor·OPC-UA Simulator

조우진*, 여채은**, 구재회***, 임채영****

Woojin Cho*, Chea-eun Yeo**, Jae-Hoi Gu***, Chae-Young Lim****

요약 본 논문은 공장에너지관리시스템에 필수적인 데이터 수집 시스템을 엣지 기반 경량화 플랫폼에서 최적으로 구축하고자 분석 및 평가한다. 실증 중인 제조 공장의 센서를 기반으로 시뮬레이션 하여 센서/OPC-UA 시뮬레이터를 개발하였으며, 개발한 시뮬레이터를 통해 엣지 디바이스의 스토리지 엔진을 평가한다. 엣지 디바이스에서 스토리지 엔진에 따른 성능을 평가하여 최적의 스토리지 엔진을 제시한다. 실험 결과 스토리지 엔진을 RoccksDB로 사용하였을 때 InnoDB를 사용하였을 때에 비해 절반 이하의 메모리와 데이터베이스 크기를 지니며 3.01배 빠른 소요시간을 지니는 것을 알 수 있다. 이 연구는 한정된 자원을 사용하는 디바이스에서 시계열 데이터를 관리할 때 유리한 스토리지 엔진을 선택할 수 있으며, 센서/OPC 시뮬레이터를 통한 해당 분야 추가 연구에 기여한다.

주요어 : 데이터베이스, 엣지 디바이스, 에너지 관리 시스템, FEMS, LSM-tree, MyRocks, MySQL, OPC-UA, PLC, RDBMS, 센서, 시뮬레이터

Abstract This paper analyzes and evaluates to optimally build a data collection system essential for factory energy management systems on an edge-based lightweight platform. A "Sensor/OPC-UA simulator" was developed based on sensors in an actual food factory and used to evaluate the storage engine of edge devices. The performance of storage engines in edge devices was evaluated to suggest the optimal storage engine. The experimental results show that when using the RocksDB storage engine, it has less than half the memory and database size compared to using InnoDB, and has a 3.01 times faster processing time. This study enables the selection of advantageous storage engines for managing time-series data on devices with limited resources and contributes to further research in this field through the sensor/OPC simulator.

Key words : Database, Edge device, Energy Management System, FEMS, LSM-tree, MyRocks, MySQL, OPC-UA, PLC, RDBMS, Sensor, Simulator

*정회원, 고등기술연구원 에너지환경IT융합그룹 연구원 (제1저자) Received: April 14, 2023 / Revised: April 30, 2023

**정회원, 고등기술연구원 에너지환경IT융합그룹 연구원 (참여저자) Accepted: May 8, 2023

정회원, 고등기술연구원 에너지환경IT융합그룹 연구위원 (참여저자) **Corresponding Author: cylim@iae.re.kr

****정회원, 고등기술연구원 에너지환경IT융합그룹 선임연구원 (교신저자) Dept. of Energy Environment IT Convergence Group,

접수일: 2023년 4월 14일, 수정완료일: 2023년 4월 30일 Institute for Advanced Engineering, Korea

게재확정일: 2023년 5월 8일

I. 서론

전 세계적으로 에너지 소비 절감을 위해 기존의 경험적, 감각적인 에너지 관리 방식이 아닌 데이터, 지능형, 능동형 에너지 관리 방식을 차용하고자 지속적으로 발전하고 있다 [1][2].

이러한 에너지 관리 측면에서 이를 시스템화 하여 EMS라고 하는 에너지 관리 시스템을 구축하여 데이터 관리나 에너지의 정보 확인, 관리, 장비 제어 등을 통한 에너지 절감에 대해 관심이 증대되고 있다 [3].

에너지 관리 시스템은 적용되는 곳에 따라 빌딩 에너지 관리 시스템(BEMS), 주택 에너지 관리 시스템(HEMS), 공장 에너지 관리 시스템(FEMS) 등으로 나누게 된다. 이 중 에너지 관리 시스템에서 가장 큰 비중을 차지하는 공장 에너지 관리 시스템(FEMS)은 제 3차 에너지기본계획을 통해 2025년부터 10만TOE 이상의 에너지 다소비 사업장을 대상으로 의무화가 될 것이다.

이러한 에너지 관리 시스템(EMS)의 핵심적인 기술을 구현하기 위해서는 “데이터 수집 시스템 구축”이 가장 선행되어야 한다. 에너지 관리를 위한 장비의 예지 보전, 에너지 예측, 절감 방식 제안 등의 기능을 수행하기 위해서는 데이터가 존재해야 하고, 데이터를 수집하기 위해서는 데이터 수집 시스템이 기초적으로 구축되어 있어야 하기 때문이다 [4].

하지만 지금까지의 공장의 데이터 수집 방식은 전통적인 데이터 수집 방식인 수기 작성 혹은 HMI(Human Machine Interface)나 SCADA(Supervisory Control and Data Acquisition)를 통해 수집되었다. 허나 수기 작성의 경우 인력의 한계로 인해 기입 가능한 몇몇의 정보만 수집하는 것이 대부분이었다. 또한 HMI는 초창기의 경우 데이터를 단순히 출력을 시키는 것에 치중되어있었으며 SCADA의 경우 고비용 및 스토리지 효율 및 최적화 등의 측면에서 데이터 수집에 최적화 되어있지 않아 데이터 수집에는 한계를 지니고 있다.

이러한 문제를 해결하기 위해 PLC(Programmable Logic Controller) 데이터를 실시간으로 수집할 수 있는 데이터베이스 시스템 구축이 필요하다. 하지만 에너지 효율에 있어 기존의 서버는 문제가 있으며, 현장의 상황은 열악하고, 한정된 공간만을 지니고 있어 대형 서버 랙을 구비하기 어려운 경우가 많고, 구비를 한다 하여도 환경적 측면에 있어 제대로 된 관리가 어려워 관리적 측면

과 공간 제약성이 유리한 소형 엣지 디바이스를 이용하는 것에 대한 고려가 필요하다.

하지만 이러한 엣지 디바이스는 보편적으로 낮은 성능을 지니기 때문에 데이터베이스의 선택이 매우 중요하다. 이 때 흔히 센서 데이터와 같은 시계열 데이터를 수집할 때 선택하는 시계열 데이터베이스의 경우 굉장히 제한된 사용성만을 제공할 뿐 아니라 ARM 기반의 임베디드 디바이스에 적합하지 않거나 지원하지 않는 경우가 많다. 따라서 배포성 및 이식성과 사용성이 우수하며 사용처에 따라 스토리지 엔진을 선택하여 성능을 개선할 수 있는 오픈소스 데이터베이스인 MariaDB를 선택하여 다양한 스토리지 엔진을 평가하여 엣지 디바이스에서의 센서 데이터 수집에 가장 적합한 스토리지 엔진을 찾고자 한다.

본 논문에서는 실제 제조 공장의 환경을 모사하여 온도, 압력, 유량등의 센서에서 수집되는 데이터를 시뮬레이션하는 센서/OPC-UA(Open Platform Communications Unified Architecture) 시뮬레이터를 개발하여 실제 환경과 유사한 환경을 구현하여 테스트를 진행한다. 센서 데이터를 제너레이션 한 데이터를 OPC UA를 통해 엣지 디바이스의 데이터베이스에 적재할 시 데이터베이스의 스토리지 엔진마다 CPU 사용량, 메모리 사용량, 데이터베이스의 크기, 소요 시간 등을 평가하여 스마트 팩토리 혹은 에너지 관리 시스템에 적합한 스토리지 엔진을 분석하고 평가한다.

본 논문의 실험 결과로 시뮬레이터 기반 실험에서는 I/O에 의한 병목이 없어 RocksDB의 경우 더 높은 CPU 활용율과 절반 이하의 RAM사용량을 보였으며, 두 배 이상 작은 데이터베이스 크기를 지니는 것을 알 수 있다. 또한 대량 적재(Bulk Insert) 상황에서는 RocksDB를 스토리지 엔진으로 선택한 것이 InnoDB를 사용한 것에 비해 소요시간은 3배 가량 성능이 우수했으며, 램 사용량은 절반 정도를 사용하는 것을 확인할 수 있다.

II. 센서·OPC-UA 시뮬레이터

1. 시뮬레이션

엣지 디바이스에서 에너지 관리 시스템을 위한 데이터 수집 플랫폼을 평가하기 위해선 필연적으로 시뮬레이터가 필요하다. 기존의 벤치마크인 TPC-C와 같은 OLTP(Online Transaction Processing) 벤치마크들의 경

우 온라인 banking이나 쇼핑, 주문 입력등을 통해 “동시에 발생”하는 다수의 트랜잭션을 처리하는 즉, 단위 시간당 다수의 쿼리를 수행하는 것에 목적을 둔다. 따라서 단위 시간당 트랜잭션의 처리량, 쿼리의 처리량을 보는 TPS(Transaction per second)나 QPS(Query per second)를 측정하는 것에 특화 되어있다. 하지만, 센서 데이터 수집은 샘플링 하는 시간에 따라 x라는 단위 시간당 구축된 센서의 개수인 N개의 센서 데이터를 “Insert”만 하는 것을 주목적으로 한다. 따라서 그림 1의 기존의 초당 몇 개의 쿼리를 처리 할 수 있는지에 대한 평가가 그림 1의 단위 시간당 하나의 데이터를 저장하는 엣지 디바이스 데이터베이스의 평가에서는 주된 평가 요소가 아니다.

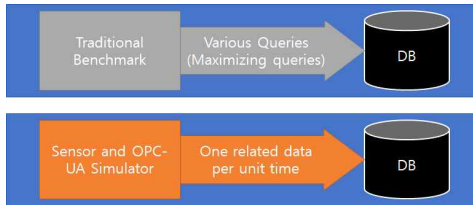


그림 1. 기존 벤치마크와의 차이
 Figure 1. Differences from existing benchmarks

또한, 센서 데이터의 경우 누적의 경우 지속적으로 쌓이고, 온도의 경우 1초만에 10°C가 움직이지 않는 것과 같이 기존의 값에 의존적이게 되는데 기존의 벤치마크는 이에 대한 고려가 없기 때문에 이와 같은 특성을 활용하는 데이터베이스에서는 제대로 된 성능 측정이 불가능하다.

따라서 본 논문에서는 기존의 단순한 서버, 기기의 성능 혹은 데이터베이스의 성능 평가가 아닌 “에너지 관리 시스템을 위한 데이터 수집 플랫폼” 특성에 적합한 시뮬레이션 기법이 필요하게 된다.

그에 따라 본 논문에서는 사양이 낮은 엣지 디바이스에 데이터 수집에 대한 평가를 위해 기존의 벤치마크만을 활용하지 않고, 실제 환경과 센서 데이터의 데이터 연관성을 모사하여 시계열로 연관성이 있는 데이터를 전송할 시 발생하는 부하를 보는 것으로 엣지 디바이스의 스토리지 엔진의 평가를 진행한다.

2. 센서·OPC-UA 시뮬레이터

실제 실증하고 있는 제조 공장의 센서는 온도, 압력(순시, 누적), 유량등의 센서가 있다는 것을 확인하고 해

당 센서를 기반으로 온도, 압력(순시, 누적), 유량 등을 모사하여 실제 환경에 보다 유사한 시스템을 구현하였다. 각 센서는 그 전 값에 의존성이 있게 발생시키도록 하였다. 누적 값이 아닌 값의 경우 최대, 최저값 (예시. 온도 : -10~35)을 지니고 조정하였다.

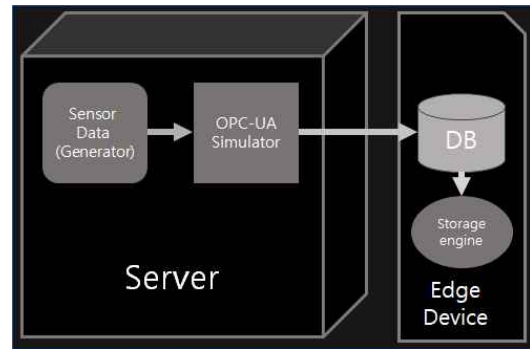


그림 2. 센서/OPC-UA 시뮬레이터 구조
 Figure 2. Sensor and OPC-UA Simulator Structure

그림 2와 같이 컴퓨팅 과정 및 시뮬레이션 과정은 엣지 디바이스에서는 데이터베이스 본연의 성능만을 비교하기 위하여 Xeon Silver 서버에서 Ubuntu 20.04에서 구동하였고, 시뮬레이터는 Python을 사용하여 구현해 이더넷 통신을 통해 데이터베이스에 데이터를 전달하도록 개발하였다.

III. 데이터베이스

1. 스토리지 엔진

그림 3과 같이 MySQL은 용도에 따라 저장을 수행하는 스토리지 엔진을 플러그인 형태로 사용할 수 있는 구조를 지니고 있다.

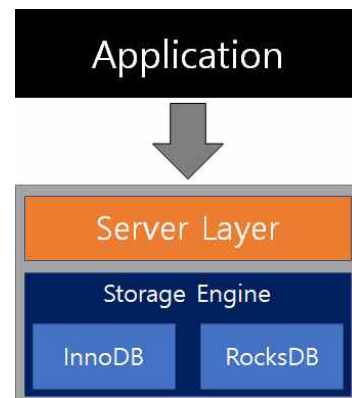


그림 3. MySQL 구조 (Storage engine 구조)
 Figure 3. MySQL structure (Storage engine structure)

기존의 전통적인 관계형 데이터베이스에서 보편적으

로 사용되는 B-Tree 기반의 스토리지 엔진의 경우 인덱싱 방식에 의해 조회 성능이 매우 뛰어나지만 지속적인 in-place update에 의해 쓰기 시 높은 메모리 사용량과 CPU 사용량에 의해 성능이 하락하는 문제점을 지니고 있다. 그에 따라 보편적으로 대다수의 관계형 데이터베이스가 쓰기 집약적인 형태에서는 사용에 불리하기에 NoSQL, 시계열 데이터베이스등이 그 대안으로 사용되었다. 하지만, NoSQL 데이터베이스와 시계열 데이터베이스들은 높은 운용 난이도와 데이터를 수집만 하는 것이 아닌 다양한 용도로 활용하게 되는 현재에는 수집한 데이터를 통한 조회 작업에서의 손해와 편의 기능 및 특수 기능 지원의 부재로 인해 NoSQL 데이터베이스는 결과적으로 관계형 데이터베이스와 함께 사용하게 되었다. 또한, 엣지 디바이스와 같이 ARM CPU를 사용하는 디바이스에 대한 지원 및 저장 공간의 제약에 의한 활용을 제대로 하지 못하는 문제점이 존재한다.

본 논문에서는 특수 목적성을 띄는 경우 적합한 엣지 디바이스가 지속적인 데이터 수집을 할 때의 평가를 하고자 ARM 디바이스에서도 안정적으로 동작하며, 확장성 및 이식성 등에 이점이 있는 MariaDB를 사용하여 실험을 진행했다. 센서/OPC-UA 시뮬레이터의 특성상 쓰기 집약적 작업이기 때문에 특정 스토리지 엔진에서는 실시간(1초당 N개의 센서 데이터를 수집)으로 데이터를 수집하는 상황에서 엣지 디바이스의 낮은 성능과 메모리의 제약으로 인해 스토리지 엔진에 따른 차이가 있을 것으로 예상하여 설계하였다.

본 논문에서는 스토리지 엔진의 특성에 따라 엣지 디바이스에서 센서 및 IoT 데이터 수집에 적합한 스토리지 엔진을 비교하여 스토리지 엔진의 특성을 분석하고 적합성을 보여주고자 한다.

표 1. 스토리지 엔진 사양
Table 1. Storage engine Spec

	Index	SQL	압축	트랜잭션
InnoDB	지원	지원	지원	지원
RocksDB	지원	지원	지원	지원
Archive	미지원	일부지원	지원	미지원
CSV	미지원	지원	미지원	미지원

표 1은 주 데이터베이스 서버에서 자주 사용되는 스토리지 엔진인 InnoDB와 RocksDB, 백업 데이터베이스 서버에서 사용될 수 있는 Archive와 CSV 엔진의 사양이다.

우리는 위의 스토리지 엔진 중 실생활에서 가장 보편적으로 사용되고 있는 MySQL의 기본 스토리지 엔진인 InnoDB와 Facebook에서 개발한 MyRocks의 스토리지 엔진인 RocksDB를 비교하고자 한다 [5].

InnoDB의 경우 B-tree 구조를 지닌 전통적인 관계형 데이터베이스를 위한 스토리지 엔진이다 [6]. 반면 RocksDB는 LSM-tree라는 구조를 지닌 NoSQL중 Key-value store로써 순차 쓰기와 적은 메모리에서 보다 유리한 요건을 지닌 Key-value store이다. 각각의 스토리지 엔진의 데이터 수집 시 엣지 디바이스에서의 성능 평가를 진행한다.

2. Parameter Setting

본 논문의 실험에서 사용할 엣지 디바이스인 Raspberry Pi 4 model b의 경우 낮은 성능으로 인해 InnoDB와 RocksDB의 기본 설정으로 사용하기에는 어려움이 따른다.

따라서 MySQL의 Config 파일을 수정하여 표 2와 같은 설정으로 실험을 진행한다.

표 2. 데이터베이스 변수 설정
Table 2. Database parameter setting

InnoDB	buffer_pool=2GB, key_buffer_size=128MB
RocksDB	write_buffer_size=512MB, write_buffer_number=4 Compression = LZ4 & ZSTD

IV. 실험 및 결과

1. 실험 환경

실험을 위해 사용한 엣지 디바이스는 라즈베리 파이 4 model B이며, 그 사양은 표 3과 같다 [7].

표 3. 엣지 디바이스 사양
Table 3. Edge Device Spec

Model	Raspberry Pi 4 model B
CPU	ARM v7
RAM	4GB
Storage	Samsung Evo Plus Class 10 128GB mSD
OS	Ubuntu 20.04 LTS

운영체제로는 Ubuntu 20.04 LTS를 사용하였으며, 사용한 데이터베이스는 MariaDB 10.2를 선택하였다. MariaDB 10.2에서 스토리지 엔진 InnoDB와 RocksDB

로 바꾸어가며 평가를 진행하였다.

표 4. 센서/OPC-UA 시뮬레이터 사양
 Table 4. Sensor/OPC-UA Simulator Spec

Language	Python 3.10.6
Sensor	Temperature, Pressure, Flow
CPU	Intel Xeon Silver 4210R
Memory	64GB
OS	Ubuntu 20.04 LTS

센서 데이터 제너레이터 및 시뮬레이터의 구현은 표4와 같은 사양의 서버와 Python 버전에서 FreeOPCUA를 통해 구현하였다. 스토리지 엔진의 영향을 평가하기 위한 실험이므로 스토리지 엔진과 시스템적 영향을 엮지 디바이스에 미치게 하지 않기 위해 별도의 서버에 OPC-UA 시뮬레이션 서버를 구축하였다. 이더넷 통신을 통해 OPC-UA 서버에서 데이터를 받아오는 방식을 취하였으며, 해당 데이터는 0.1초에 한 번 100개의 센서 데이터(제조 공장 센서 기준)를 받아오는 것을 모사하여 설계하였다.

2. 실험 결과

1) 시뮬레이터 기반 스토리지 엔진 성능 평가

개발한 시뮬레이터를 통해 회당 100개의 센서 데이터와 타임스탬프만 수집하도록 하였으며, 평균 CPU 사용량과 평균 RAM 사용량, 최종 DB 크기를 평가 기준으로 평가하였다. 수행한 데이터의 양은 10일치이다.

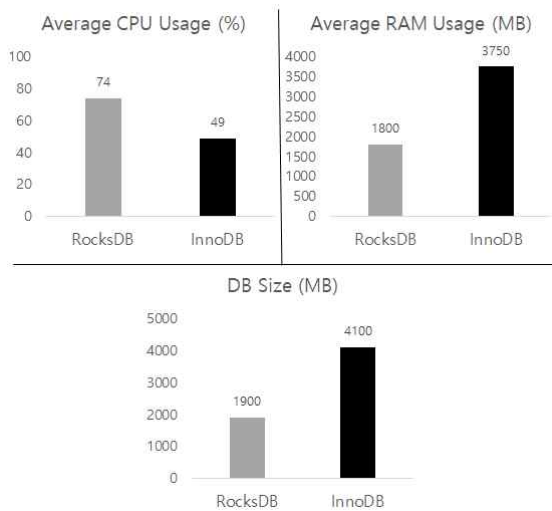


그림 4. 시뮬레이션 기반 실험을 통한 스토리지 엔진의 성능 평가 그래프

Figure 4. Performance evaluation graph of storage engine through simulation-based experimentation

그 결과는 그림 4와 같이 나오는 것을 볼 수 있었다. 우선 RAM 사용량이 InnoDB가 RocksDB에 비해 2배 이상 높은 것을 볼 수 있는데, 이는 InnoDB의 Buffer Pool에 의해 발생한다. 쓰기 작업 시 메모리 부하가 큰 것을 알 수 있다. 반면 RocksDB의 경우 LSM tree의 강점으로 인해 Write buffer에 수집된 데이터를 순차 데이터로 바꾸어 flush만 하면 되기 때문에 메모리 사용량이 더 적은 것을 볼 수 있다.

또한 InnoDB의 경우 In-place 업데이트를 위해 실제 데이터가 있는 공간만큼만 사용하는 것이 아니라 8KB, 16KB와 같이 예약된 공간을 할애하는 압축 방식 때문에 발생하는 오버헤드로 인해 압축이 일정 크기 이하로는 불가능하여 데이터의 압축 효율이 좋지 않다. 또한, RocksDB 스토리지 엔진의 경우 델타 압축이라는 기법을 통해 데이터 간의 차이를 이용하여 데이터를 압축하기 때문에 데이터 압축에 유리하며, 그 외에도 예약된 공간이 없는 SST file 형식등 데이터의 크기를 줄이는 여러 기법들을 통해 데이터베이스의 크기를 줄일 수 있어 InnoDB에 비해 두 배 이상 적은 용량을 사용하는 것을 볼 수 있다.

마지막으로 CPU 사용량의 경우 오히려 RocksDB의 경우가 더 높은 것을 볼 수 있는데, 이는 InnoDB의 경우 I/O가 병목이 되어 오히려 낮은 CPU 사용량을 기록하는 것을 볼 수 있다. RocksDB의 경우 일반적인 경우 CPU 사용량이 60%대를 유지하였으나, Compaction과 같은 고부하 작업의 발생 시 90%까지 상승하여 높은 CPU 사용량을 기록하였다.

2) 대량 적재(Bulk Insert)

데이터베이스를 복제하거나, 백업 데이터베이스로 주기적인 백업을 사용할 때와 같이 특정한 상황에서 대량의 데이터 적재는 필수적으로 사용된다. 특히 센서 데이터 수집과 같은 데이터를 수집하는 데이터베이스의 경우 시간에 따라 지속적으로 데이터가 쌓이기 때문에 보편적인 시계열 데이터베이스들의 경우 일정 기간 이상이 지난 특정 기간 이전의 데이터를 자동으로 삭제하도록 지원한다. 따라서 주 데이터베이스에서는 해당 데이터들이 삭제되기 때문에 그 이전에 사용자의 경우 해당 기간에 따라 데이터를 백업하면 불필요한 공간 낭비를

막을 수 있다.

따라서 대량 적재 실험을 하기 위하여 본 논문에서는 기본키는 시간으로 이루어지고, 60개의 실수형 센서 데이터로 이루어진 실제로 실증 중인 제조 공장의 데이터셋을 12GB 가량 추출하여 사용하였다.

해당 평가에서는 데이터의 적재 소요 시간과 적재 시 CPU 사용량 및 RAM 사용량, DB 크기까지 복합적인 요소를 고려하여 테스트를 진행한다.

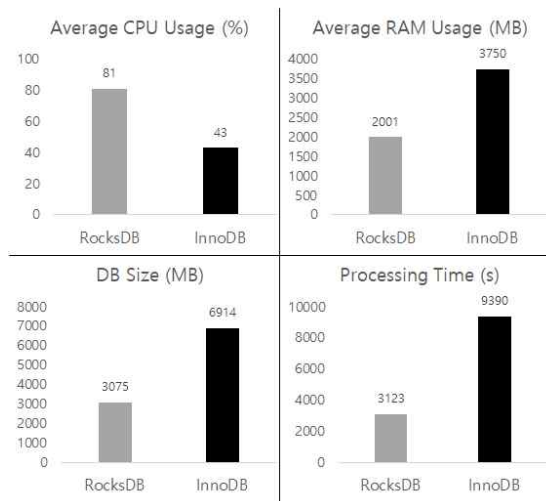


그림 5. 대량 적재 실험을 통한 스토리지 엔진의 성능 평가 그래프

Figure 5. Performance evaluation graph of storage engine through bulk insert experimentation

결과로 그림 5와 같이 대략 3.01배 가량 InnoDB가 RocksDB 스토리지 엔진을 사용하였을 때보다 소요시간이 오래 걸리며 RAM Usage의 경우도 여전히 InnoDB가 많이 사용하는 것을 볼 수 있다. 하지만 메모리 사용량의 격차가 줄어든 것을 볼 수 있는데, 이는 더 단시간에 더 많은 데이터의 입력으로 인해 Compaction이 기존보다 많이 발생하여 단시간에 더 잦은 Compaction의 호출로 인해 평균이 상승되었다. 또한 이 잦은 Compaction에 의해 앞서서보다 더 큰 CPU 사용량의 차이를 발생시켰다. InnoDB의 경우 CPU의 성능보다 과한 쓰기 발생으로 인해 I/O가 병목이 되어 오히려 기존의 센서/OPC 시뮬레이터를 사용할 때보다 CPU 사용량이 낮아졌다.

시뮬레이터를 구동하였을 때와 같은 원인으로 데이터베이스의 크기 또한 차이가 나는데, 시뮬레이터 기반 스토리지 엔진 성능 평가와 달리 보다 InnoDB와

RocksDB의 차이가 큰 이유는 RocksDB 스토리지 엔진의 경우 Bulk Insert 시 맨 아래 레벨에 데이터를 모두 Insert하게 되며, 레벨 별 압축 알고리즘을 선택 가능한 특성에 따라 맨 아래 레벨의 압축 알고리즘은 ZSTD, 나머지 레벨에서는 속도가 중요한 LZ4로 압축하여 맨 아래 레벨로 적재되는 Bulk insert 시 보다 높은 압축률을 보여 InnoDB의 압축률은 큰 차이를 보이지 않지만 RocksDB의 압축률이 보다 높아져 발생하는 차이이다.

V. 결론

본 논문에서는 엣지 디바이스에서 효과적이고 효율적인 스토리지 엔진에 대해 평가하였다. 그를 평가하기 위하여 센서 데이터 모사와 OPC-UA 시뮬레이션을 통해 시뮬레이터를 제작하여 실제 환경과 흡사한 환경에서 엣지 디바이스의 평가를 진행하였다. 그 결과 InnoDB에 비해 RocksDB가 삽입 집약적인 작업에서 소요시간은 3.01배, DB 크기에서는 2.16배 가량 이득이 있었다. 따라서 시계열 데이터베이스를 사용하지 않을 정도의 데이터 양을 입력해야 할 시 MySQL을 사용할 때에는 스토리지 엔진을 RocksDB로 설정하는 것이 보다 큰 이득을 얻을 수 있음을 알 수 있다.

현재 시뮬레이터의 센서 데이터 제너레이터를 개선하여 온도의 경우 계절성, 압력의 경우 실제 데이터를 기반으로 가동 시간에 따라 구동하는 방식과 같이 시뮬레이션의 고도화를 이룩하고자 하며 OPC-UA 시뮬레이터의 경우에도 여러 PLC 시뮬레이션 및 100개의 센서 데이터를 고정하고 실험하는 것이 아닌 유동적으로 변경 가도록 고도화 시키고자 한다.

향후 제작한 시뮬레이터를 개선하여 해당 연구를 통해 엣지 디바이스에서 시계열 데이터베이스와 관계형 데이터베이스, NoSQL 데이터베이스의 성능 비교를 통해 엣지 디바이스에서 최적의 데이터베이스를 제시하고 저사양 디바이스에 맞춰 데이터 인덱싱 등의 방식을 개선하는 연구를 진행하고자 한다.

References

[1] S. H. Moon, "Big Data Platform Construction and Application for Smart City Development," The journal of the convergence on culture technology,

- vol. 6, no. 2, pp. 529 - 534, May 2020.
DOI: <https://doi.org/10.17703/JCCT.2020.6.2.529>
- [2] Chae-Young Lim, Chae-Eun Yeo, Hyung-Ah Lee, Sung-Yul An, Jae-Hoi Gu.(2022).Design and Implementation of IoE System for Energy Consumption Platforms.Proceedings of Symposium of the Korean Institute of communications and Information Sciences,(),449-450.
- [3] Kim, C. W., Kim, J., Kim, S. M., & Hwang, H. T. (2015). Technological trends and case studies of factory energy management systems (FEMS) for energy saving in manufacturing industries. Equipment Journal, 44(1), 22-27.
- [4] Gi-Seok Lee and Sang-Hyun Lee. "A Study on Real-time Detection Using Odor Data Based on Mixed Neural Network of CNN and LSTM" The International Journal of Advanced Culture Technology 11, no.1 (2023) : 325-331.
- [5] Matsunobu, Yoshinori, Siying Dong, and Herman Lee. "MyRocks: LSM-tree database storage engine serving facebook's social graph." Proceedings of the VLDB Endowment 13.12 (2020): 3217-3230.
DOI: <https://dl.acm.org/doi/10.14778/3415478.3415546>
- [6] MariaDB, "InnoDB", <https://mariadb.com/kb/en/innodb/>
- [7] 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-8gb>

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