복합 시계열의 유연한 처리를 위한 고유 식별자 기반 시계열 데이터 연산처리 엔진

오승민, 지영민, 권동우

한국전자기술연구원

{osm1892, ym.ji, dwkwon}@keti.re.kr

A Unique Identifier-Based Engine For Flexible Processing of Composite Time-Series Data

Sheungmin Oh, Youngmin Ji, Dongwoo Kwon

Korea Electronics Technology Institute (KETI)

요 약

다양한 산업 현장에서 실시간 시계열 데이터 분석의 중요성이 커지고 있으나, 기존 시계열 데이터베이스는 복잡한 사용자 정의 연산 처리에 한계를 가진다. 본 논문은 이러한 문제를 해결하기 위해 고유 식별자를 기반으로 다양한 기본 연산, 통계 처리, 사용자 정의 함수 등을 포함하는 사용자 정의 연산을 유연하게 실행하고 관리할 수 있는 고성능 시계열 데이터 연산처리 엔진을 제안한다. 제안 시스템은 REST API, 도메인 별 데이터 입출력 모듈, 메인 모듈로 구성되어 기존 시스템과의 손쉬운 통합 및 확장이 가능하다. 본 시스템을 통해, 사용자는 기반 시스템 변경 없이도 다양한 시계열 데이터 간 복잡한 연산을 쉽게 처리함으로써 복합적인 분석 요구사항에 효과적으로 대응할 수 있다.

Ⅰ. 서 론

산업 현장, 스마트 빌딩, IoT 환경 등 다양한 응용 분야에서 실시간 시계열 데이터 분석의 중요성이 지속적으로 증가하고 있다. 특히 공장 에너지 관리 시스템(Factory Energy Management System, FEMS)[1]은 실시간으로 방대한 양의 계측 데이터를 수집하고 분석해야 하므로, 고성능의 데이터 처리 및 연산 능력이 필수적이다.

기존의 시계열 데이터베이스(Time-Series Database, TSDB)는 대량의 데이터 저장과 기본 조회 기능은 효과적으로 지원한다. 그러나 여러 시계열 간의 복잡한 사용자 정의 연산을 직접적으로 처리하는 데에는 한계가 존재한다. 이러한 한계를 극복하기 위해 Apache Spark[2]과 같은 분산 처리 프레임워크가 활용될 수 있으나, 이는 복잡한 클러스터 환경 구축 및 유지보수에 상당한 자원을 소모한다. 또한, Prometheus[3] 등 일부 TSDB는 고도화된 시계열 연산 쿼리를 지원하지만, 일반적인 외부 TSDB 환경에서 적용하기에는 적합하지 않다.

이러한 문제를 해결하기 위해서는 시스템 규모에 따라 효율적으로 확장할 수 있는 스케일링을 지원하고, 기존 시스템과의 통합 운용이 가능한 시계열 연산 시스템이 요구된다.

또한, 다양하고 복잡한 데이터 처리 작업을 간편화하기 위해서는 기본 연산뿐 아니라 데이터들의 통계 연산 및 AI 분석 등의 사용자 정의 함수를 사용할 수 있어야 하며, 메트릭과 태그의 조합을 기반으로 데이터를 일관된 형식으로 표현하는 고유 식별자를 기반으로 연산 작업을 명확하게 정의하고, 필요에 따라 유연하게 연산을 추가, 삭제, 감시, 관리할 수 있는 기능이 필요하다.

본 논문에서는 TSDB 메트릭과 태그 정보를 조합하여 생성되는 도메인별 고유 식별자를 기반으로, 사용자의 다양한 데이터 연산 요청에 따라 주기적으로 TSDB의 시계열 데이터 간 연산을 유연하게 처리하고, 연산 과정을 효과적으로 관리할 수 있는 고성능 시계열 데이터 자동처리 엔진을 제안한다.

Ⅱ. 본론

본 논문에서 제안하는 시계열 연산 엔진은 그림 1과 같이 REST 기반 요청 처리 API, 인메모리 KV 저장소, 도메인별 데이터 입출력 모듈, 메인 모듈로 구성된다.

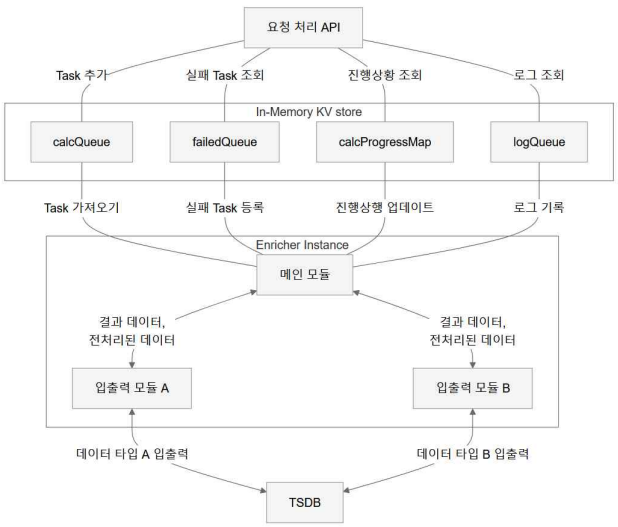

그림 1. 시스템 구조

요청 처리 API는 데이터 연산 요청을 받아 주기적으로 calcQueue에 연산 요청을 추가하는 작업을 수행한다. 이를 통해 사용자는 고유 연산자를 기반으로 새로운 연산을 손쉽게 정의하고 시스템에 등록할 수 있다. 또한, 현재 실패한 작업과 작업별 진행 상황, 데이터 처리 과정에서 기록되는 로그를 실시간으로 조회할 수 있도록 하였다. 연산 요청 형식은 표 1과 같이 수식, 데이터 도메인(데이터 타입), (1m-ago)와 같은 TSDB 질의 호환 형

식의 상대 시간 질의 범위, 스케줄링 형식 및 주기, 최대 재시도 횟수로 구성된다. 스케줄링 형식은 (5 * * * * *)과 같은 스케줄링 구성이 가능한 cron 타입과 (5 min)과 같은 구성이 가능한 interval 형식 중 하나를 선택할 수 있다.

표 1. 연산 요청 형식

항목	설명	예시
수식	연산 요청용 수식	KFEMS.A.00.KW = KFEMS.A.01.KW + KFEMS.A.02.KW
데이터 타입	데이터 도메인	fems
질의 시작시간	요청 기준 질의 시작시간	5m-ago
질의 종료시간	요청 기준 질의 종료시간	1m-ago
스케줄링 형식	cron/interval 중 사용할 형식	interval
cron	cron 호환 스케줄 형식	5 * * * *
interval	시간 간격 기반 스케줄 형식	5m
재시도 횟수	데이터 질의 실패 시 재시도 횟수	3

도메인별 데이터 입출력 모듈은 각 도메인의 데이터 형식에 맞춰 데이터의 질의, 변환, 전송을 수행하는 플러그인 형식의 모듈이다. 각 도메인은 서로 다른 TSDB, 고유 식별자 규칙, 사용자 정의 함수를 가질 수 있다. 따라서 이러한 다양한 경우를 메인 로직에서 대응하는 것이 어렵기 때문에, TSDB 데이터로부터 이러한 식별자를 생성하는 로직을 데이터 입출력 모듈에서 처리함으로써 메인 모듈은 일관된 형식으로 데이터 연산을 처리할 수 있는 유연성을 확보하였다.

메인 모듈은 인메모리 KV 저장소의 calcQueue에서 개별 작업을 가져와, 수식을 SQL 기반으로 토큰화하고, 작업에 정의된 도메인 형식에 따라서 각 플러그인을 호출하여 데이터를 질의하고, 수식을 바탕으로 DuckDB[4]를 이용해 데이터 간 연산을 수행하여 TSDB에 전송하는 일련의 작업을 관리한다. 수식 토큰화 과정에서는 PostgreSQL과 DuckDB에서 사용되는 libpg_query 라이브러리를 사용함으로써 안정적이고 효과적으로 토큰화를 수행할 수 있도록 하였다.

DuckDB는 오픈소스 임베디드 온라인 분석 처리(Online Analytical Processing, OLAP) 데이터베이스로, 단일 바이너리 혹은 라이브러리 형태로 사용 가능하다. DuckDB는 지연 연산과 쿼리 최적화를 지원하기 때문에 효율적인 연산이 가능하며, SQL 연산을 기반으로 하기에 다른 RDBMS와 비슷한 방식으로 데이터 분석을 수행할 수 있다. 또한, Apache Arrow[6]를 지원하므로 이와 호환되는 다양한 데이터프레임을 이용한 연산을 수행할 수 있다.

인메모리 KV 저장소는 모든 데이터를 메모리상에 저장하여, 키에 따라 빠르게 값을 저장 및 조회할 수 있는 데이터베이스이다. 다른 데이터베이스와 달리 데이터가 HDD와 같은 저장소를 경유하지 않으므로 매우 빠르게 여러 프로세스, 시스템 간에 데이터를 주고받을 수 있어, 이를 기반으로 고성능의 실시간 데이터 처리가 가능하다. 대표적인 예시로 Redis가 존재한다.[5]

본 논문에서 제안하는 연산 처리 시스템의 성능을 측정하기 위하여 5초 간격의 데이터로 구성되어 1일 당 약 17000 포인트의 시계열 데이터를 포함하는 1, 3, 7, 15, 30일 범위의 TSDB 데이터를 이용하여 데이터 처리 성능을 측정하였다. 측정 방식은 파일 시스템으로부터 데이터를 읽는 과정을 포함하여, 데이터 전처리와 5개의 시계열 데이터 간 연산을 수행하는 과정을 포함하여 전체 소요 시간을 측정하는 방식이다. 측정 결과, 데이터 크기에 따라 총 처리 시간이 선형적으로 증가하는 것을 확인하였으며, 30일 범위의 데이터를 처리하는 데 약 13초가 소요되는 것을 볼 때 대량의 데이터를 합리적인 시간 내에 처리할 수 있음을 알 수 있다.

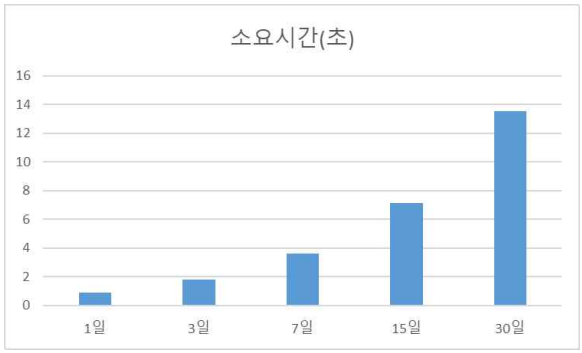

그림 2. 시스템 연산 성능 측정

Ⅲ. 결론

본 논문에서는 DuckDB를 기반으로 하여 사용자의 요청에 따라 복합 시계열 데이터 연산을 자동으로 처리하는 시스템을 제안하였다. 제안된 시스템은 기존 시스템과 쉬운 통합이 가능하며, 다양한 규모의 시스템에 설치하여 원활한 동작을 수행할 수 있다. 이를 통해 기반 시스템을 변경하지 않고도 다양한 복합 시계열 처리가 가능하다. 추후 연구를 통해 다양한 사용자 정의 함수를 적용하여 더 다양하고 복잡한 연산 수요를 충족하며, 대규모의 시계열 연산에 대응 가능하도록 하는 방안을 연구할 예정이다.

ACKNOWLEDGMENT

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (RS-2023-00237018)

참 고 문 헌

[1] D. Lee and C.-C. Cheng, “Energy savings by energy management systems: A review,” Renewable and Sustainable Energy Reviews, vol. 56, pp. 760 - 777, Apr. 2016, doi: 10.1016/j.rser.2015.11.067.

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud '10), Boston, MA, USA, Jun. 2010, pp. 1 - 7.

[3] P. B.C., H. Maddirala, and S. M., "Implementing an Effective Infrastructure Monitoring Solution with Prometheus and Grafana," International Journal of Computer Applications, vol. 186, no. 38, pp. 7-15, Sep. 2024.

[4] M. Raasveldt and H. Mühleisen, "DuckDB: an Embeddable Analytical Database," in Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19), Jun. 2019.

[5] H. Bang, S.-H. Kim, and S. Jeon, “Comparative Evaluation of Data Processing Performance between MySQL and Redis,” Journal of Internet Computing and Services, vol. 25, no. 3, pp. 35 - 41, Jun. 2024.

[6] A. Lamb, Y. Shen, D. Heres, J. Chakraborty, M. O. Kabak, L.-C. Hsieh, and C. Sun, "Apache Arrow DataFusion: A Fast, Embeddable, Modular Analytic Query Engine," in Companion of the 2024 International Conference on Management of Data (SIGMOD '24), Santiago AA, Chile, 2024, pp. 5-17, doi: 10.1145/3626246.3653368.