Critical 서비스를 위한 데이터 레이크하우스를 활용한 데이터 분석

전효건, 김종원*
광주과학기술원 지스트 대학 전기전자컴퓨터전공, *광주과학기술원 AI 대학원
gyrjs6537@gist.ac.kr, *jongwon@smartx.kr

Data Analytics employing Data Lakehouse for Critical Services

HyoGeon Jeon, JongWon Kim*
Gwangju Institute of Science and Technology(GIST) College School of Electrical Engineering and Computer Science(EECS), *GIST AI Graduate School.

요 약

본 논문에서는 Cloud-native Edge Computing 환경에 Data Lakehouse 를 도입하여 Data Lakehouse 의 Open Table 을 이용할 경우 Data Lake 에 비교해 빠른 분석이 가능함을 제안하고 이에 대하여 간단하게 검증한다.

Ⅰ. 서 론

최근 실시간 빅데이터 분석 과정의 중요성과 고가용성, 확장성 등을 고려한 Cloud-native Edge Computing 환경이 늘어나고 있다. 많은 기업들은 Cloud-native Edge Computing 에서 데이터를 실시간으로 분석하기 위하여 데이터를 수집, 전처리, 분석과 조치 그리고 데이터 저장의 네 단계를 거친다. 특히 데이터 저장 단계에서 저비용의 다양한 형태의 데이터를 저장 가능한 Data Lake 를 이용한다[1].

이러한 Data Lake 는 수집된 데이터를 가공하지 않은 상태로 저장소에 로드하고 이후 필요시 가공하는 과정을 거친다. 하지만 데이터를 명확한 기준 없이 Data Lake 에 저장할 경우 데이터에 접근이 어려운 상태가 될 수 있다. 이를 보완하고자 등장한 개념이 Data Lakehouse 로 Data Lake 와 Data Warehouse 의 장점을 더하여 등장한 개념이다. 이러한 Data Lakehouse 의 경우 모든 형태의 데이터를 Data Warehouse 의 정형 데이터처럼 취급할 수 있다는 장점이 있다[1].

본 논문에서는 Cloud-native Edge Computing 환경에 Data Lakehouse 를 도입하여 데이터 분석을 진행할 경우 Critical Service 에 대응할 가능성이 있음을 순차적으로 보일 것이다.

Ⅱ. Data Lakehouse 의 강점: Open Table

본문에서는 서론에서 언급한 Cloud-native Edge Computing 에서 Data Lakehouse 를 이용한 데이터 분석과 Data Lake 를 이용하였을 경우의 차이를 보겠다. 우선 Data Lake 와 Data Lakehouse 의 차이를 기술적 관점으로 살펴보면 Data Lake 는 원시 데이터 자체를 관리하는데 비해 Data Lakehouse 는 Open Table 을 이용한다[2]. 이 Open Table 의 역할을 살펴보기 위하여 현재 가장 많이 이용되는 Open Table 들을 살펴보았다.

Delta Lake	Iceberg	Hudi
		

표 1. 대표적인 Open Table 3 종류의 형식 비교

위의 표 1 과 같이 살펴본 결과 Open Table 형식은 실제 데이터를 압축률이 높고 I/O 가 빠른 형식으로 저장하고 이를 메타데이터 혹은 로그를 이용하여 관리함을 알 수 있다[3].

더 나아가 Open Table 이 Data Lakehouse 에서만 지원하는 기능을 Open Table 중 하나인 Delta Lake 를 이용하여 확인하였다. 그 결과 Data Lake 는 데이터를 새로 저장할 때 파일을 교체하지만, Delta Lake 를 이용하면 데이터를 Delta 형식으로 변경 후 기존의 데이터에 추가하는 방식으로 저장한다. 따라서 Data Lake 에서 지원하지 않는 전처리를 Data Lakehouse 에서는 Open Table 을 이용하여 구현할 수 있다.

따라서 다음 본문부터 위의 Data Lakehouse 의 Open Table 의 특성을 이용하여 Cloud-native Edge Computing 에서 데이터 분석을 진행한다면 데이터 수집 후 데이터가 최종으로 이용되기 전 까지 발생하는 과정을 단축시켜 결과적으로 Critical Service 에 대응할 수 있음을 보일 것이다.

Ⅲ. Cloud-native Edge Computing 에서 Data Lakehouse 를 이용한 데이터 분석 환경 설계 및 구현

앞선 본문 Ⅱ의 결론을 검증하기 위하여 우선 Cloud-native Edge Computing 에서 Data Lakehouse 를 이용한 데이터 분석 환경을 설계하였다. Data Lakehouse 는 Data Lake 로부터 등장하였으므로 이를 먼저 설계하였다. 본 논문에서는 모든 형태의 데이터의 저장이 용이하고 확장이 쉬운 Object Storage MinIO 와 다중 규모의 데이터 처리 및 관리를 하는 Apache Spark 를 이용하여 Data Lake 를 설계하였다. 앞서 언급된 모든 프레임워크는 다중 규모의 계산과 많은 컴퓨터 자원의 필요함에 따라 Cloud-native 하게 설계하였다[4].

다음으로 설계된 Data Lake 를 바탕으로 Data Lakehouse 를 이용한 데이터 분석 환경을 설계하겠다. 우선 Data Lakehouse 의 경우 Apache Spark 와 완전히 호환되는 Delta Lake 의 Delta 형식을 이용하였다. 또한 전반적인 데이터 분석을 위한 데이터 수집 및 분석

단계가 필요하였다. 이를 반영하여 Data Lakehouse 를 Cloud-native 하게 설계하였다.

구현상 특이사항으로 MinIO 의 경우 쿠버네티스 버전 1.19 이상을 요구하므로 1.26 버전을 이용하였다. 쿠버네티스를 이용하여 설계에 따라 Cloud-native 하게 데이터 분석 환경을 구현하였다.

Ⅳ. Cloud-native 한 Data Lakehouse 를 이용한 데이터 분석 과정 검증

본 논문의 가설을 검증하기 위해 Cloud-native 한 환경에서 기존에 많이 사용되던 Data Lake 를 이용한 데이터 분석과 Data Lakehouse 를 이용한 분석을 비교하였다. 가설에 따라 데이터 수집 및 조치 과정 이후는 동일하다고 가정한 뒤 수집된 데이터를 데이터 분석까지 처리되는 과정을 비교하였다. 우선 검증을 위하여 GPS 데이터가 필요한 상황에서 수집된 데이터는 GPS 를 포함한 다양한 차량 관련 데이터라고 가정하였다. 시나리오에 따라 임의의 차량 정보를 Spark 에서 JSON 형식으로 생성하였다.

우선 Data Lake 를 이용한 데이터 분석 환경에서는 생성된 JSON 데이터가 MinIO 에 저장되어 있음을 확인할 수 있다. 이후 GPS 분석을 위해 데이터를 이용하였을 때 처음 수집한 형태 그대로 원시 상태였으므로 분석을 위해 전처리를 따로 진행해야 했다.

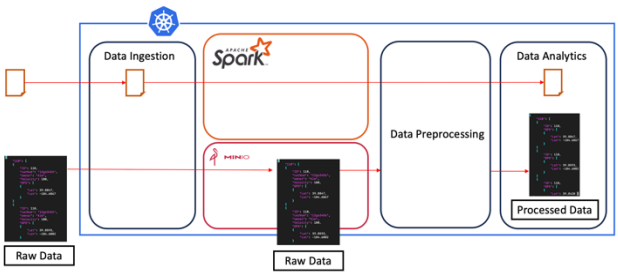

그림 1. 구현된 Cloud-native Data Lake 에서 데이터 분석 시 데이터가 변화하는 과정

다음으로 Data Lakehouse 를 이용한 데이터 분석 환경에서는 생성된 JSON 데이터가 parquet 형식으로 변환되고 JSON 형태의 로그가 생성되어 Delta Table 의 형식으로 저장되어 있음을 확인할 수 있다. 구현된 Data Lakehouse 에서 Delta 형식을 이용하여 저장된 상태에서 전처리가 가능하였다. 이후 GPS 분석을 위해 데이터를 이용하였을 때 가공된 데이터를 바로 사용할 수 있었다.

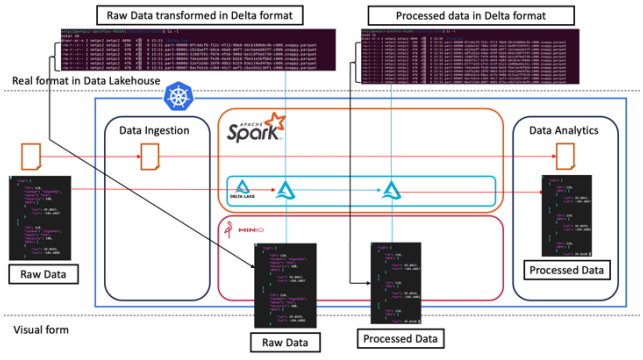

그림 2. 구현된 Cloud-native Data Lakehouse 에서 데이터 분석 시 데이터 및 Delta 형식이 변화하는 과정

V. 결론

본 논문의 검증 과정을 통하여 Edge Computing 의 데이터 분석 환경에서 Cloud-native Data Lakehouse 를 이용하였을 경우 Cloud-native Data Lake 를 이용하였을 경우에 비해 Open Table 을 이용하여 데이터 저장과 동시에 데이터 전처리 과정을 진행하여 데이터 처리 과정을 단축시키는 것을 확인하였다. 이는 데이터 분석 시 즉시 데이터를 이용할 수 있게 하므로 기존의 데이터 전처리 과정에서 발생하던 지연을 감소시킬 수 있다. 따라서 Cloud-native Edge Computing 에서 Data Lakehouse 를 이용하였을 경우 Data Lake 를 이용하였을 경우보다 정형 데이터를 이용한 실시간 분석이 필요한 Critical Service 에 더 적합하다.

수요가 증가하는 실시간 데이터 처리 서비스와 발전된 IoT 기술로 인하여 Critical Service 에 대한 중요성이 증가하고 있다. 본 논문의 Edge Computing 에서 Cloud-native 한 Data Lakehouse 를 이용한 데이터 분석의 규모를 확장한 뒤 실제 다양한 데이터를 통한 검증이 있다면 초저지연의 데이터 분석 환경이 등장 가능함을 기대할 수 있다. 이를 적용하여 서론에서 언급하였던 스마트 시티, 자율주행 자동차 등 실시간 데이터 처리 기술이 필요한 분야에서 적절하게 활용될 수 있길 희망해본다.

ACKNOWLEDGMENT

이 논문은 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신산업진흥원의 지원(No.S0101-23-1002, 악천후 등 외부환경 대응 가능한 V2X 기반 connected 플랫폼 기술 개발)과 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No. 2019-0-01842, 인공지능대학원지원(광주과학기술원))을 받아 수행된 연구임.

참 고 문 헌

[1] HyoGeon Jeon and JongWon Kim, “Designing Data Analytics employing Data Lakehouse for Critical Services Realization”, 2023

[2] Michael Armbrust, “Lakehouse: A New Generation of Open Platform that Unify Data W arehousing and Advanced Analytics”, 2021

[3] Vohra. D, Apache Parquet, In: Practical Hadoop Ecosystem. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2199-0_8, 2016

[4] JaeMyung Song and JongWon Kim, “Design of Cloud-native V2X Edge Computing Environment employing Data Lakehouse”, 2022