In-memory 기반 Programmatic 쿼리 프로세싱 방안 제안 및 성능 분석

김지해, 이상범
SK 브로드밴드

wlgo6020@sk.com, sb.lee@sk.com

A Proposal and Performance Analysis of In-memory based Programmatic Query Processing Scheme

Kim Jihae, Lee Sangbeom
SK Broadband

요 약

본 논문은 빅데이터를 효과적으로 사용하기 위해 기존의 DBMS 에서 NoSQL 로 데이터를 적재하는 아키텍처를 소개하고, 이를 위해 필요한 SQL 을 프로그래밍 기법을 활용해 처리 하는 방법을 제안한다. 제안한 방법은 기존의 단일 SQL 방법의 성능을 측정했을 때, 4.72 배 정도 뛰어난 속도를 보여 주였다.

Ⅰ. 서 론

빅데이터 저장을 위해서는 다양한 유형의 데이터를 저장 및 관리하기 위한 시스템이 필요하다. DBMS 에서는 스키마를 기반으로 하는 정형화된 데이터(structured data)를 관리하였으나, 이는 빅데이터에서 생성되는 다양한 유형의 데이터를 저장하기에는 적합하지 않다. 또한 기존의 관계형 데이터베이스에서 정규화 원칙에 따라 테이블들이 세세히 나누어져 있기 때문에 사용자가 원하는 데이터를 추출하기 위하려면 다중 조인 및 Sub Query 가 들어간 SQL 을 작성할 수 밖에 없다. 이처럼 데이터베이스의 복잡도가 높아지고 있을 뿐만 아니라 대량의 데이터 조회 시 저성능 이슈 또한 동반하기 때문에 NoSQL 이라는 대안이 떠오르고 있다.

NoSQL 중에서도 키-값 저장소 (key-value store)는 모든 데이터를 키(key)와 값 (value)의 쌍으로 저장함으로써 임의의 유형의 데이터를 저장할 수 있는 유연한 데이터 모델을 지원한다. 이러한 특징 덕분에 현재 NoSQL 은 IoT, 웹 시스템, 추천 시스템. 검색 시스템, 데이터 분석 등의 다양한 분야에서 활용되고 있다.[1-5]. 따라서 본 논문에서는 기존의 데이터베이스가 아닌 NoSQL 을 시스템에 활용하고자 하며, 그를 위해 SQL 에서 조인 및 데이터 처리를 하는 것이 아니라, 프로그래밍을 통해 처리하는 방안을 제안한다.

Ⅱ. 본론

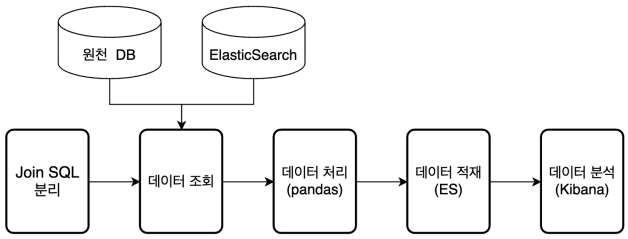
키-값 저장소로는 Elasticsearch 를 선택하였으며, Elastic 스택 중 하나이자 시각툴 인 Kibana 를 활용하여 데이터 분석을 용이하게 하고자 한다.[6]

[표 1] 사용 환경

DB	Opensource NoSQL	Language	Library
Oracle	Elasticsearch 7.9.3 Kibana 7.9.3	Python 3.6.8	Pandas 1.1.5 Elasticsearch 6.3.1

[표 1]는 사용된 시스템과 환경을 기재하였다. 사용하는 DB 는 Oracle 이며, OpenSource NoSQL 인 Elasticsearch7.9.3 과 Kibana 7.9.3 을 사용하였다. 가벼운 언어로 빠르게 처리하기 위하여 Python 의 Pandas Library 를 선택하였다. Pandas 는 Python 프로그래밍 언어를 기반으로 구축된 오픈 소스 데이터 분석 및 조작 Library 로서 빠르고 유연하며 사용하기 쉬운 특징 덕분에 머신러닝, 딥러닝에 특히 많이 쓰인다[7].

제안하고자 하는 방법은 기존에 있던 DB 에서 데이터를 추출 후 적재하는 것이 아니라 DB 데이터와 NoSQL 데이터를 다 사용하여 다양한 데이터를 생성 및 처리 후 적재를 하도록 한다. 프로그래밍을 통해 이러한 처리를 훨씬 유연하게 할 수 있으며 성능 또한 뛰어나다고 할 수 있다.



[그림 1] 제안 방법 흐름도

[그림 1]은 본 논문에서 제안하는 프로그램 단계를 보여준다. 첫 번째로 기존에 쓰이는 복잡한 쿼리를 분리한다. 본 실험에서는 111 줄의 하나의 SQL 문을 5 개의 SQL 문으로 분리하였다. 기존의 방법과 달리 조인을 SQL 에서가 아닌 Python 에서 처리하는 것이다. 두 번째로 각각의 SQL 문을 DB 에서 조회하고 추가로 필요한 데이터를 Elasticsearch 에서 필요한 데이터 또한 조회한다. 세 번째로는 조회한 데이터들을 파이썬 Pandas 를 통해 조인 및 필드 추가 등의 데이터 처리를 하도록 한다. 네 번째로는 최종 데이터를 파이썬 라이브러리를 통해 Elasticsearch 에 저장한다. 마지막으로 시각화 툴인 Kibana 를 통해 데이터 추출 및 분석을 하도록 한다[2]. DB 에서 Elasticsearch 간의 싱크를 준 실시간성으로 하기 위해 DB 에 무리가 가지 않는 선인 30 분마다 프로그램을 실행시키도록 한다.

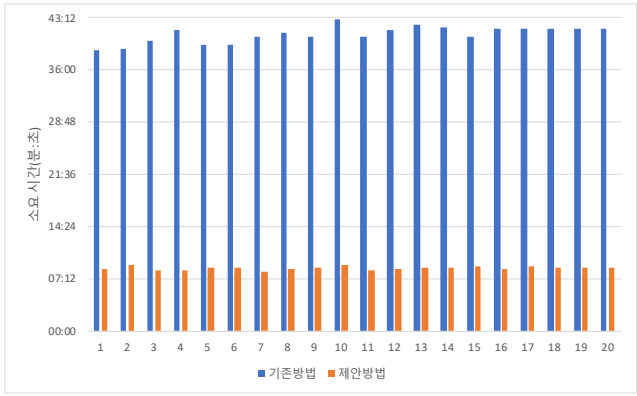
제안한 프로그램의 성능을 기존의 SQL 문을 썼을 때와 비교해보았다. [표 2]는 실험에 사용한 데이터의 특성을 보여준다. 총 데이터 개수는 49 만건이며, 총 데이터 크기는 232.7MB 이다.

실험 환경은 다음과 같다. 운영체제는 CentOS Linux release 7.6.1810 (Core), CPU 는 4 Cores Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz, 메모리는 8GB, 디스크는 200GB SSD 를 사용하였다. 동일한 조건에서 실험을 진행하기 위해 모두 Python 을 사용하였고 같은 Elasticsearch 에 적재하였다.

기존 방법과 제안 방법 모두 python 에서 데이터 조회, 처리, 적재 단계를 거친 후 시간을 측정하였다. [그림 2]는 총 20 번의 테스트를 진행한 결과를 보여준다. 전체적으로 제안 방법이 우수한 성능을 보였다. 기존 방법은 평균 40 분 56 초의 시간이 걸렸고, 제안 방법은 평균 8 분 40 초가 걸렸다. 제안 방법이 4.72 배 정도 빠르게 처리한 것을 확인할 수 있다. 이를 OPS(Operation per Second)로 변환했을 때 기존 방법은 초당 200 건의 데이터를 처리 할 수 있으며 제안 방법은 초당 947 건의 데이터를 처리 할 수 있다.

[표 2] 사용된 데이터 특성 요약

데이터 총 개수	데이터 컬럼 수	총 크기	도큐먼트 평균 크기
492,267 건	34 개	232.7MB	952Byte

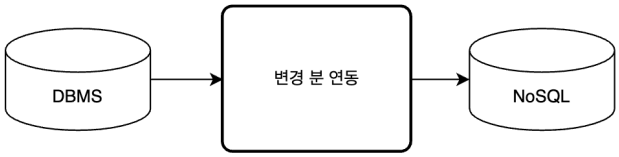


[그림 2] 실험 결과

Ⅲ. 결론

본 논문에서는 Python, Elastic 스택을 활용하여 DBMS 의 데이터를 빠르게 적재 및 활용하는 방법을 제안하였다. 제안된 방법은 기존의 SQL 문 방법과 비교했을 때 약 4.72 배의 우수한 성능을 보였다.

이는 준 실시간성을 띄고 있어서 실시간을 필요로 하는 서비스에 적용하기는 어려움이 있다. 그림은 DBMS 에서 바로 NoSQL 로 변경분만 적재하는 설계도이다. 추후에는 DBMS 에서 변경 분 만 NoSQL 에 적재하여 실시간 성을 보장하는 연구가 진행될 필요가 있다.



[그림 3] DB 와 NoSQL 간의 직접 연동

참 고 문 헌

[1] 홍성삼, 김동욱, 홍성표, 김병곤, 이재강. (2020). 도로포장 품질관리에 적합한 NoSQL 기반의 IoT 빅데이터 고속 수집 및 분석 시스템. 대한공간정보학회지, 28(4), 99-107.

[2] 김정준, 곽광진, 박정민. (2019). 미세먼지 분석 서비스를 위한 NoSQL 기반 센서 웹 시스템. , 19(2), 119-125.

[3] 고은정, 김호준, 박효주, 전영호, 이기훈, 신사임. (2017). NoSQL 기반 연관 콘텐츠 추천 시스템의 설계 및 구현. 멀티미디어학회논문지, 20(9), 1541-1550.

[4] 심형남, 김정동, 백두권. (2011). 실시간 검색을 위한 NoSQL 기반의 TK-인덱싱 기법. 한국정보과학회 학술발표논문집, 38(2C), 77-78.

[5] 윤도현, 이종혁, 김응모. (2014). NoSQL 기반 드라마 유사성 분석. 한국인터넷정보학회 학술발표대회 논문집, (), 331-332.

[6] https://www.elastic.co/kr/kibana

[7] https://pandas.pydata.org