실시간 발전소 시설 장비 센서 데이터에 대한 빅데이터 스트리밍 질의 처리 시스템 설계 및 구현

엄정호*, 유찬희*, **, Komal Sarda*, **, 박경석*, **
*한국과학기술정보연구원
**UST 빅데이터과학과
jhum@kisti.re.kr, rbyche@kisti.re.kr, komalsarda@kisti.re.kr, gspark@kisti.re.kr

Design and Implementation of Big Data Streaming Query Processing System for Realtime Power Plant Sensor data

Jung-Ho Um*, Chan Hee Yu*, **, Komal Sarda*, **, Kyongseok Park*, **
*Korea Institute of Science and Technology Information
**Department of Big Data Science, UST

요 약

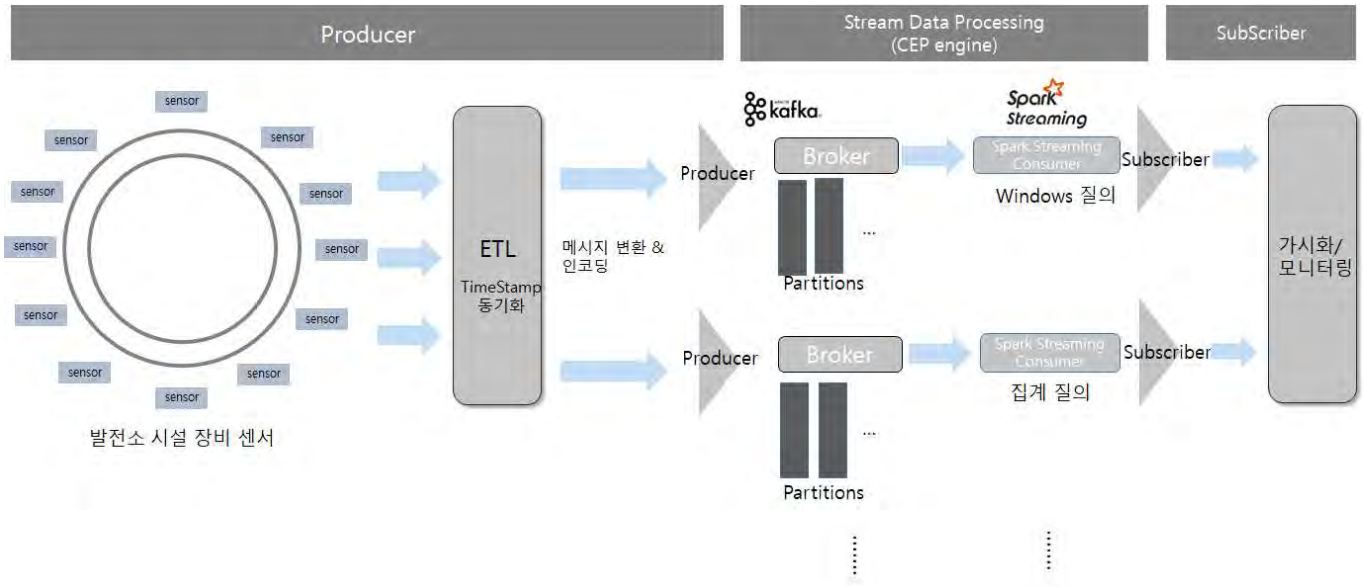
발전 시설은 연간 무중단으로 운영되어야 하고, 고장이 발생하면 손해가 막대하기 때문에 발전 시설 장비에는 수십만 개의 센서 데이터가 설치되어 있다. 본 논문에서는 효율적인 센서 데이터의 수집과 시설 모니터링 및 고장 예측 등을 위한 빅데이터 스트리밍 질의 처리 시스템을 설계 및 구현하였다. 또한 실시간 데이터 수집의 효율적인 관리를 위해 인코딩 방식을 설계하였으며, 데이터 전송 성능을 측정하여 문자열로 데이터를 전송하는 것보다 평균 12%, 최대 32% 데이터 처리 성능이 향상됨을 보였다. 또한, 스트리밍 데이터에 대한 윈도우 질의 처리 성능을 측정하여 약 0.97초의 평균 집계 질의 처리 시간이 소요됨을 확인하였다. 향후에는 고장 감지를 위한 인공지능 추론 모델을 제안하는 빅데이터 스트리밍 질의 처리 시스템에 적용할 예정이다.

1. 서론

발전 시설은 연간 무중단으로 운영되어야 한다. 만일 발전 시설 장비가 고장이 발생하면, 그 손해는 막대하기 때문에 발전 시설에는 장비의 상태를 모니터링하기 위한 수십만 개의 센서가 설치되어 있고, 실시간으로 데이터를 수집하고 있다. 실시간으로 수집되는 데이터는 빅데이터 스트리밍 질의 처리와 분석을 통해 고장 예측이나 상태 이상을 감지한다. 빅데이터 처리 시스템은 크게 수집과 분석으로 나눌 수 있으며, 스트리밍 데이터 수집에서는 카프카(Kafka)[1], 플럼(Flume)[2] 등의 오픈 소스가, 스트리밍 데이터 분석에는 스톰(Storm)[3], 스파크 스트리밍(Spark Streaming)[4]등의 오픈 소스가 활발하게 활용되고 있다. 본 논문에서는 이러한 수십만 개의 센서로부터 생산된 데이터를 실시간으로 수집하고, 빅데이터 스트리밍 질의 처리를 하기 위한 시스템을 설계 및 구현한다. 아울러, 실시간 센서 데이터의 수집 성능은 빅데이터 스트리밍 질의 처리를 위해 중요한 요소이기 때문에 제안하는 시스템에서 센서 데이터의 효율적인 저장 관리를 위한 인코딩 방식을 설계 및 성능을 측정하여 효율적임을 보였다.

2. 관련 연구

일반적으로 빅데이터 실시간 데이터 처리는 복합 이벤트 처리(Complex Event Processing:CEP) 형태[5]을 활용한다. CEP 시스템은 크게 실시간 데이터를 생산하는 Producer, 실시간 데이터를 처리하는 이벤트 처리 엔진, 처리된 데이터를 입수하는 Subscriber 구조로 되어 있다. 제안하는 시스템 또한 이러한 구조에 따라, Producer 에 해당하는 부분은 카프카를 활용하였으며, 이벤트 처리 엔진으로는



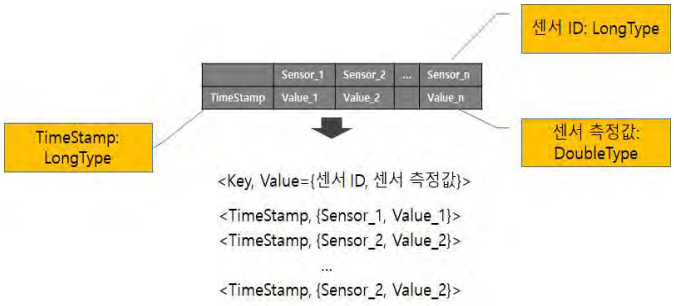
(그림 1) 전체 시스템 개념도

스파크 스트리밍 컴포넌트를 활용하였다. 카프카는 비정형 데이터를 <Key, Value> 형태로 관리하며, 주로 실시간 데이터를 메모리에 저장했다가 실시간 데이터 분석 시스템에 전달하는 역할을 한다. 카프카는 데이터를 전송하는 프로듀서(Producer), 데이터를 메모리로 구성된 Partition에 저장 및 서비스하는 브로커(Broker), 데이터를 브로커로부터 전달받아 스트리밍 데이터 처리 또는 스트리밍 데이터 분석에 활용하는 컨슈머(Consumer)로 구성된다. 한편, 스파크는 분산 메모리 기반으로 데이터를 추상화한 RDD(Resilent Distributed DataSet)로 데이터를 관리한다. 스파크 스트리밍은 RDD로 표현되는 데이터 셋에 대하여 빅데이터 스트리밍 질의 처리를 지원한다. 대표적으로Sliding Window, Aggregation 질의가 있다. 스파크 스트리밍에서는 마이크로 배치(Micro Batch) 형태로 질의 처리를 지원한다. 스파크 스트리밍을 통해, LSTM[6] 과 같은 시계열 인공지능 모델 기반으로 발전소 시설 장비 고장 예측 모델을 수행하기에 적합한 환경을 제공할 수 있다. 스파크 스트리밍의 마이크로 배치 처리 방식은 센서 데이터를 일정 시간 동안 모아서 스트리밍 질의를 처리하기에, 이러한 고장 예측 추론 질의 처리에 적합하다.

2. 시스템 구조

발전 시설에 대한 센서 데이터의 수집과 빅데이터 스트리밍 질의 처리를 위한 전체 시스템 구조는 (그림 1)과 같다.

시스템은 일반적인 실시간 복합 이벤트 처리 시스템[5]의 구조에 따라 구성하면 다음과 같다. 데이터를 생산하는 컴포넌트는 발전소 시설 장비에서 측정되는 센서 데이터를 타임스탬프(TimeStamp)별로 동기화(그림 1의 ETL(Extract, Transform, Load) 서버)하여, 센서 데이터를 아래 (그림 2) 와 같은 메시지 구조로 인코딩한 후, 카프카의 브로커 서버에게 전송한다.



(그림 2) 메시지 인코딩 구조

메시지 구조는 그림 2와 같이 TimeStamp를 Key로 할당하며, 센서 ID와 센서 측정값을 하나의 Value로 그룹핑하여 인코딩한다. TimeStamp와 센서 ID는 Long Type으로 저장을, 센서 측정값은 Double Type의 자료형으로 저장한다. Value의 자료형은 ByteBuffer를 활용하였다. 따라서 Key에는 4 바이트, Value에는 12 바이트가 할당된다. 센서 측정값의 정확도를 낮추면 4 바이트의 float 형으로도 저장 가능하다.

브로커에 인코딩된 센서 데이터는 스파크 스트리밍 라이브러리에서 제공하는 카프카 컨슈머[7]를 통해

스파크 스트리밍 프로세스로 입수된다. 데이터 분석 및 처리하는 파트인 스파크 스트리밍 컴포넌트에서는 윈도우 질의, 집계 질의 등의 빅데이터 스트리밍 질의 처리를 통해 외부 가시화 시스템에 질의 결과를 전달할 수 있다.

3. 성능 평가

제안한 실시간 센서 데이터에 대한 인코딩 구조의 효율성을 측정하기 위해, 문자열(String Type)의 구조로 데이터를 전송할 경우와 인코딩을 수행한 후 데이터를 전송했을 때의 시간과 Throughput을 측정하였다. 성능평가를 위해 발전소에서 수집한 약 4일 분량의 센서 데이터를 <표 1>과 같이 데이터 볼륨별로 분할하였다. 아날로그 101,740 개의 센서를 1초 단위로 동기화하여 이를 한 행으로 하는 데이터를 사용하였으며, 카프카의 프로듀서는 csv 형태로 저장된 데이터를 읽어서 브로커에 전달한다.

<표 1> 센서 데이터의 정보

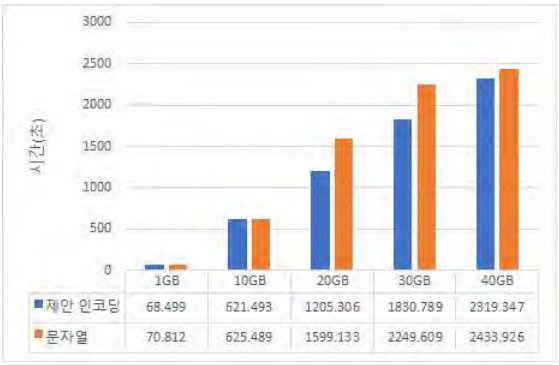
Volume(GB)	건수(행/초)	측정 시간
1	8,709	2시간 25분 9초
10	87,177	24시간 11분 57초
20	174,243	48시간 24분 3초
30	261,361	72시간 36분 1초
40	342,003	95시간 0분 3초

성능평가에 활용한 카프카 서버는 총 3대이며, 각 서버마다의 하드웨어 사양은 <표 2>와 같다.

<표 2> 서버 하드웨어 사양

CPU	Intel Xeon E5-2660 2.6GHz*20core
Memory	128GB
HDD	7200RPM SAS*6TB
Network	FDR 56Gbps

성능평가에 활용한 카프카는 Confluent 5.5.0에 포함되는 카프카를 활용하였으며, 3.0 버전의 스파크를 활용하였다. 카프카에는 서버별로 총 3개의 브로커를 할당하였으며, Zookeeper는 1대의 서버에 할당하였다. 카프카에서 replication_factor 는 1 로, 파티션의 개수는 2로 설정했다.

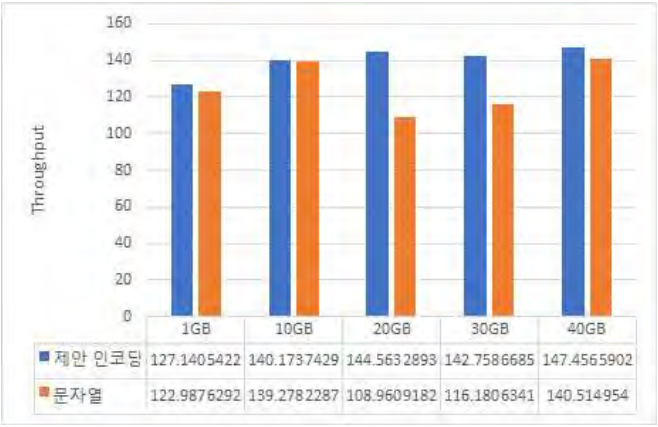


(그림 3) 데이터 전송 시간 성능 비교

제안하는 메시지 인코딩 구조와 문자열 형태로 수집 데이터를 카프카의 프로듀서를 통해 데이터를 읽어서 인코딩 방식에 따라 메시지 포맷을 변환하여, 카프카의 브로커에 전송하는 시간을 측정한 결과는 (그림 3) 과 같다. 데이터의 크기가 크기 않을 때에는 시간 차이가 많지 않지만, 데이터가 클수록 제안하는 인코딩 기법이 약 200 ~300초 정도 시간을 절약하였다.

Throughput 성능 측정은 아래 식 (1) 과 같이 Throughput(T)을 정의하여, 제안하는 인코딩 기법과 기존 문자열 방식을 (그림 4)와 같이 비교하였다.

T=데이터 전송 시간/센서 데이터 측정 시간 (1)



(그림 4) Throughput 성능 비교

(그림 4)에서와 같이 제안하는 인코딩 방식이 Throughput의 성능을 약 12% ~ 약 32%까지 향상시켰다.

한편, 스파크의 빅데이터 스트리밍 질의 처리 성능 측정을 위해, 한 노드에서 10초 간격으로 지난 20초 간의 카프카에서 입수된 모든 센서 데이터에 대하여 평균을 각 센서 ID 별로 계산하는 질의 수행 시간을 측정하였다. 1분 동안 총 6번의 질의에 총 65.829초

가 소요되었다. 질의 처리 시간만을 계산하면 평균 약 0.9715 초 정도 소요되었으며, 이는 20초간의 모든 데이터에 대하여 스파크의 맵-리듀스를 통해 각 센서별로 평균을 계산하는 시간에 해당한다.

4. 결론

본 논문에서는 발전소 센서 데이터의 처리를 위해 빅데이터 스트리밍 질의 처리 시스템을 설계 및 구현하였다. 수십 만개의 센서로부터 초단위의 센서 데이터 측정값을 효율적으로 처리하기 위한 메시지 구조를 설계하였으며, 성능평가를 수행하여 효율적임을 보였다. 또한, 스파크 스트리밍을 활용하여 스트리밍 질의가 잘 처리됨을 확인하였다. 향후에는 스파크를 분산 환경에 적용하여 스트리밍 질의 처리 성능을 평가하고, 고장 예측을 위한 인공지능 추론 모델을 빅데이터 스트리밍 질의 처리 시스템에서 실행하는 환경을 구축할 예정이다.

사사

이 논문은 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구입니다. (No. 20181110100420)

참고문헌

[1] Jay Kreps, Neha Narkhede, Jun Rao, Kafka: a Distributed Messaging System for Log Processing, NetDB workshop '11, 2011.

[2] Flume, https://flume.apache.org/

[3] Storm, https://storm.apache.org/

[4] Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, Ion Stoica, Discretized Streams: Fault-Tolerant Streaming Computation at Scale, SOSP, Farmington, Pennsylvania, USA, 2013.

[5] G. Cugola and A. Margara, Processing Flows of Information: From Data Stream to Complex Event Processing, ACM Computing Survey, Vol. 44, No. 3, 2012.

[6] Sepp Hochreiter and Jurgen Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735 - 1780, November 1997.

[7] Spark-kafka Integration, https://spark.apache.org/docs/3.0.0-preview/streaming-kafka-0-10-integration.html