

## 대용량 실시간 데이터 처리기술에 대한 적용사례 연구

박준우\*, 정남준\*, 이인태\*  
한국전력공사\*

### A Casestudy of data processing technic in real time

Jun-Woo Park\*, Nam-Joon Jung\*, In-Tae Lee\*  
KEPCO\*

**Abstract** - 발전소와 같은 대형 플랜트에서는 많은 설비의 계측과 모니터링이 필요하다. 현장에서 계측된 다양한 데이터를 실시간으로 처리하기 위해서는 기존의 DB관리방식이 아닌 하둡 기반의 데이터 관리가 필요하며 네트워크의 속도 및 안정성도 확보되어야 한다. 클라우드와 병렬처리 방법을 통한 컴퓨팅 성능의 향상과 통신 네트워크 기술의 발전으로 200MW 화력발전소를 대상으로 한 실시간 데이터의 취득 및 처리가 가능해졌으며, 한전 전력연구원에서는 최근 필리핀 CEBU 화력발전소를 대상으로 실증시험을 시행하였다. 본 논문에서는 발전소를 대상으로 대용량 실시간 데이터의 취득, 처리, 저장 등의 일련의 과정에 대하여 다루어보고자 한다.

#### 1. 서 론

IDPP(Intelligent Digital Power Plant) 플랫폼 발전운전 데이터관리 시스템은 필리핀 Cebu 발전소의 제어 설비로부터 실시간 대용량의 운전정보를 수집하여 전력연구원 빅데이터 플랫폼으로 데이터를 연계한다. 필리핀 CEBU 발전소에서는 초당 약 10,000개의 운전데이터가 생성되고, 이 데이터는 실시간 데이터베이스(Real Time Data Base)를 통해, 실시간 감시, 추이(Trend) 분석, 각종 계산, 보고서 생성 등에 사용되며, 누적된 이력 데이터는 Trend Chart와 알고리즘 분석에 이용하기 위해 제공한다.

#### 2. 본 론

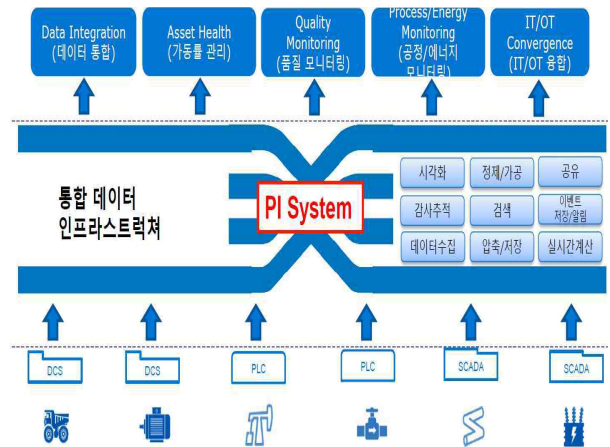
##### 2.1 시스템 구성

발전운전데이터 관리시스템은 발전설비의 운전정보를 하둡 기반의 Big Data 플랫폼과 HUBPoP 플랫폼에 기반하여 수십 Tera Byte의 데이터 관리 환경에서도 필요한 정보를 신속히 제공하고 분석할 수 있는 환경을 제공한다. 발전운전 데이터는 데이터를 분산파일 시스템에 관리하여 성능을 보장하며, 성능 개선이 필요한 경우 분산처리 프로세스를 위한 서버를 추가하여 성능을 쉽게 확보할 수 있는 구조이다. 발전운전 데이터관리 시스템은 데이터 보관 및 APP에 데이터 제공을 위해 원본 데이터를 보관하며, 특정 TAG의 신속한 Trend 조회를 위해 압축 데이터를 관리한다. 이러한 데이터는 표준화된 RestAPI를 통하여 외부 시스템 및 APP에 전달 가능하다.

##### 2.2 데이터 수집

OSI soft의 PI 시스템은 다양한 데이터 소스에서 발생된 실시간 데이터의 통합 데이터 인프라 스트럭처로서 설비의 가동률과 생산품질을 향상시키고, 생산공정 및 에너지 사용량을 효율적 관리하며, IT/OT 융합을 통한 빅데이터 분석을 지원한다.

또한 발전소의 DCS, PLC, SCADA 등의 실시간 대용량 데이터를 수집, 저장, 관리하는 솔루션으로 압축/저장, 실시간 검색, 감사추적, 검색, 정제/가공, 시각화 등의 기능을 제공한다. PI어댑터는 C#언어로 개발되었고 PI AF SDK를 이용하여 PI Server로부터 데이터를 수집 및 Confluent.Kafka 라이브러리를 사용하여 Kafka로 전송한다. PI Server로부터의 데이터 수집은 PI AF에서 제공하는 PIPointList로 데이터를 수신하여 처리한다.

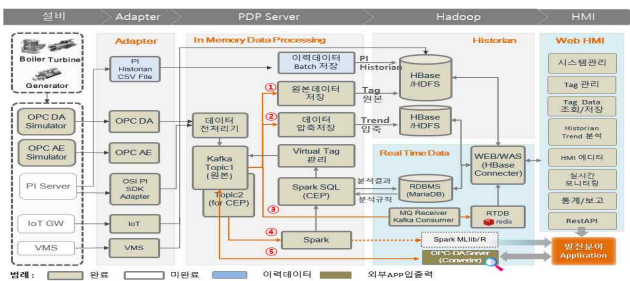


<그림 2> PI시스템 아키텍처

미국의 Capstone Technology에서 개발한 dataPARC은 기존 솔루션 PI, PHD 및 다양한 RTDB(Realtime DataBase)를 사용하는 회사의 시각화를 통합하는 시스템으로 펄프와 제지산업에 적용하기 시작하였으며, 정유, 석유화학, 화학, 전자재료, 제약 등의 다양한 산업군에서 사용되고 있다. OPC어댑터는 C#언어로 OPC DA Server/Client가 개발되었으며 GUI는 WPF로 구현하였다. 서버측은 Advosol사의 SDK를 적용하였고 클라이언트 측은 Softing사의 SDK를 적용하였다. 또한 Kafka에 전송하기 위해 Confluent.Kafka 라이브러리를 사용했다. OPC DA Server는 OPC DA 데이터를 직접 현장에서 수집할 수 없어 임의의 데이터를 Generator하여 OPC DA 데이터를 제공하는 목적으로 개발되었고, OPC DA Client는 OPC DA Server와 접속하여 서버에서 전송하는 데이터를 수집 및 Kafka로 전송하는 서비스이다.

##### 2.3 데이터 연계

Kafka Topic은 cubu 등 총 5개로 구성되어 있으며 자세한 내용은 <표 1>과 같다.



<그림 1> IDPP 플랫폼 구성도

〈표 1〉 Kafka Topic

topic 명	내 용
cebu	발전운전 데이터 수집
cebu-comp	데이터 1차압축
cebu-swing	데이터 2차압축
cebu-vms	진동데이터 수집
sys_info	주기적으로 시스템 정보를 수집

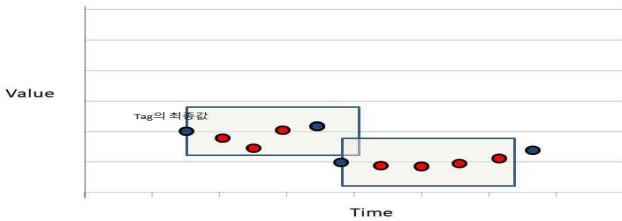
Kafka To REDIS 모듈은 Kafka Topic에 쌓인 데이터를 실시간으로 Redis에 전송하는 서비스이다. JAVA언어로 개발되었으며 Kafka에서 데이터를 받기 위해 Apache Kafka Client 라이브러리를 사용하였고 Redis에 전송하기 위해 Jedis 라이브러리를 사용하여 구현했다. Kafka 데이터 수집은 Map에 토픽별로 Kafka에서 수집된 데이터를 저장한다. Kafka To REDIS 모듈은 Kafka Topic에 쌓인 데이터를 실시간으로 Redis에 전송하는 서비스이다. JAVA언어로 개발되었으며 Kafka에서 데이터를 받기 위해 Apache Kafka Client 라이브러리를 사용하였고 Redis에 전송하기 위해 Jedis 라이브러리를 사용하여 구현했다. Kafka 데이터 수집은 Map에 토픽별로 Kafka에서 수집된 데이터를 저장한다. Redis 전송은 Map에 저장된 데이터를 Format에 맞게 수정하여 Redis에 전송한다. Kafka To HBase 모듈은 Kafka Topic에 쌓인 데이터를 이력 데이터를 저장하는 Hbase에 저장하는 기능을 수행하는 프로그램이다. JAVA언어로 개발 되었으며, 최대 대기시간(1초)이 후 구성정보 파일(kafka.properties)에 저장된 최대 데이터 개수(max.poll.records)만큼 데이터를 반복적으로 호출하여 Hbase에 저장한다.

2.4 데이터 전처리

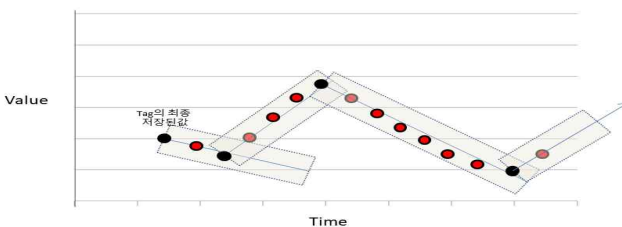
원본 데이터로부터 운전정보의 트렌드를 조회시 수 초에서 수 시간이 소요될 수 있으며, 트렌드 조회가 불가능한 상황이 발생하며, 이러한 문제를 해결하기 위해 운전데이터를 실시간으로 압축하여 조회 시간을 대폭 단축시키는 것을 목표로 한다. 특히, 조회 기간에 따라 원본, 1차 압축, 2차 압축, 3차 압축의 선별적 적용으로 신속한 데이터 조회가 가능한 것을 특징으로 한다.

Deadband 압축은 정해진 시간내 기준점으로부터 그 값이 특정 범위를 벗어났을 경우에만 데이터를 저장. 마지막에 저장된 값을 기준으로 동일한 방법으로 데이터를 압축하는 방법이다.

Swinging Door 압축은 연속된 데이터 값의 기울기에 변곡이 발생하는 시점의 데이터를 저장하여 압축하는 방식으로 Deadband의 1차 압축된 데이터를 대상으로 2차 압축하고 그 데이터는 별도의 압축데이터에 관리된다.



〈그림 3〉 Deadband 알고리즘



〈그림 4〉 Swing Door 알고리즘

2.5 데이터 저장 및 데이터 제곱

HBase 테이블 구성은 다음과 같다.

〈표 2〉 HBase 테이블 목록

table	데이터 내용	데이터 종류
idpp_tf	동시, 다수 TAG 조회 유리	발전운전
idpp_raw	특정 TAG 1개 조회 유리	발전운전
idpp_comp	deadband 적용	발전운전
idpp_swing	swingdoor 적용	발전운전
idpp_iot	iot 데이터	iot
idpp_vms	VMS 진동데이터	vms
idpp_opc-ae	OPC 알람이벤트	opc 알람,이벤트

REDIS는 Key, Value 방식으로 데이터를 관리한다. MQ Receiver는 Kafka로부터 데이터를 수집하고 HMI 및 RestAPI에서 TAG를 신속히 조회할 수 있도록 Tag명을 Key로 저장한다. 또한, HMI와 RestAPI에서 데이터 변경을 최소한으로 하기 위하여 Value 필드에 {시간, Tag, 값}의 형태로 Value를 저장한다.

Redis 구성 정보를 이용해서 Redis 서버와 Connection을 맺은 후 hgetall함수와 hmget함수를 이용해서 Redis의 데이터를 조회한다.



〈그림 5〉 Kafka - Redis 데이터 연계방안

3. 결 론

필리핀 세부 발전소를 대상으로 IDPP 플랫폼 운전, 진동, 영상정보를 취득하고 이를 원격으로 처리, 저장하여 실시간 및 이력 데이터를 API를 통해 다양한 모니터링 및 분석 애플에 연계하였으며, 발전소 BTG 및 BOP 설비를 대상으로 순시점검 및 가상고장대응훈련을 TabletPC, Mobile Phone, Hololens를 이용하여 실증하였으며 터빈분해조립, 보일러 정비, 비상발전기 기동정지 등 VR콘텐츠를 운전운을 대상으로 교육함으로써 IDPP 플랫폼 기반 ARVR 앱 서비스 실증을 완료하였다. 향후 TTA등 공신력이 있는 기관을 통해 IDPP 플랫폼에 대한 검증 및 AR/VR 서비스에 대한 개선이 필요하다. 특히 기후환경에 AR를 이용한 설비점검 서비스의 경우 민감한 설비 자동 인식과 불안정한 네트워크 환경에서도 강건한 서비스 개선이 필요하다. 발전소 디지털화의 패러다임 환경에서 필리핀 CEBU 발전소 등 국내의 발전사에 적용 시 원격 시스템 모니터링을 통한 효과적인 운영 지원 및 신속한 고장 대응을 통한 고장 처리 비용을 획기적으로 절감할 수 있을 것으로 예상된다.

[참고 문헌]

- [1] 이명호, “빅데이터 수집 처리를 위한 분산 하둡 폴스택 플랫폼의 설계”, 한국융합학회논문지, 제12권 제7호(’21.7), p.45-51
- [2] 허승범, “하둡 기반 분산 딥러닝 프레임워크에서의 스토리지 최적화 기술연구”, 명지대학교, ’21.2
- [3] 김용민 외 3명, “하둡 클러스터의 대역폭을 고려한 압축 데이터 전송 및 저장 기법”, 스마트미디어저널, 제8권 제4호(’19.12), p.46-52