

실험실정보관리시스템의 확장을 위한 오픈 소스 기반의 빅데이터 처리 기술에 관한 연구

김순곤*

A Study on Big Data Processing Technology Based on Open Source for Expansion of LIMS

Soon-Gohn Kim*

요약 실험실정보관리시스템(LIMS, Laboratory Information Management System)은 실험실 데이터를 저장, 가공, 검색 그리고 분석하기 위한 중앙화된 데이터베이스로서 검사, 분석, 시험 업무를 수행하는 실험실을 위해 특별히 고안된 컴퓨터 시스템 또는 시스템을 의미한다. 특히 LIMS는 실험실의 운영을 지원하는 기능을 갖추고 있으며, 워크플로우 관리나 데이터 추적지원 등이 필요하다. 본 논문에서는 실험실의 운영을 위하여 빅데이터 자동화 수집 기술의 하나인 크롤링 기술을 활용하여 웹사이트 및 다양한 채널에 존재하는 데이터를 수집한다. 수집된 시험 방법 및 내용 중 시험자가 활용할 수 있는 유용한 시험 방법 및 내용을 추천한다. 그리고 이에 대한 피드백을 관리하여 수집 채널의 검증이 가능한 상호보완적인 LIMS 플랫폼을 구현한다.

Abstract Laboratory Information Management System(LIMS) is a centralized database for storing, processing, retrieving, and analyzing laboratory data, and refers to a computer system or system specially designed for laboratories performing inspection, analysis, and testing tasks. In particular, LIMS is equipped with a function to support the operation of the laboratory, and it requires workflow management or data tracking support. In this paper, we collect data on websites and various channels using crawling technology, one of the automated big data collection technologies for the operation of the laboratory. Among the collected test methods and contents, useful test methods and contents useful that the tester can utilize are recommended. In addition, we implement a complementary LIMS platform capable of verifying the collection channel by managing the feedback.

Key Words : LIMS(Laboratory Information Management System), Big Data, Open Source, Web Crawling(Scraping), NoSQL/RDBMS

1. 서론

실험실정보관리시스템 (LIMS, Laboratory Information Management System)은 실험실 데이터를 저장, 가공, 검색 그리고 분석하기 위한 중앙화된 데이터베이스로서 검사, 분석, 시험 업무를 수행하는 실험실을 위해 특별

히 고안된 컴퓨터 시스템 또는 시스템을 의미한다[1]. LIMS를 통하여 실험실의 표준화되고 프로세스 고도화된 통합적인 데이터 관리가 가능하고, 분석 데이터의 수집부터 보관에 이르는 실험실의 데이터 생명주기의 관리와 결과 데이터의 재활용이 용이하다. LIMS는 정유, 석유화학, 정밀화학, 제조업, 금속, 제철, 식품, 의약, 연

This Paper was supported by Research & Development Fund of Joongbu University in 2020.

*School of Software Engineering, Joongbu University

*Corresponding Author : School of Software Engineering, Joongbu University (sgkim@joongbu.ac.kr)

Received April 06, 2021

Revised April 06, 2021

Accepted April 25, 2021

구소, 정밀화학, 보건환경, 검사소 및 기타 관련 분야를 망하하여 실험실내 시험 분석 업무에는 어느 곳이든 적용이 가능하다. 국내에서는 주로 마약·유전자감식이나 생물 정보학 관련, 미생물유전체 등에 대한 실험실정보 관리에 LIMS를 적용하고 있다[2]. Autoscribe Informatics는 BioBank, Environmental/Water, Food & Drink, Medical/Healthcare, Medical Devices 등을 비롯한 다양한 산업분야에 실험실정보관리시스템(LIMS)를 서비스하고 있다[3].

2016년 제46차 세계경제포럼(WEF) 연차총회(이하 다보스포럼)의 핵심 주제인 '제4차 산업혁명의 이해 (Mastering the Forth Industrial Revolution)'의 발표를 통하여 기술 사이의 융합으로 새로운 창조가 이뤄질 것으로 예상하고, 4차 산업혁명은 단순한 기술의 발전을 초월하여 사회전반에 혁신을 유발하고 광범위한 변화를 초래할 전망이다. 4차 산업혁명에서 이야기하는 빅데이터, 인공지능, 모바일, 클라우드, 사물인터넷, 블록체인 등이 핵심 기반기술이다[4]. 이러한 핵심 기반 기술 요소들은 제조, 서비스, 인프라 등 주요 산업분야 뿐만 아니라 실험기반의 연구를 진행하는 많은 기관에서도 다양하게 적용되어야 하며, 해당 기관들의 경우 기술혁신에 따른 트렌드 변화에 적극 대응하는 동시에 새로운 비즈니스 발굴을 위해 노력할 필요가 있다.

현대사회는 IT기술과 비즈니스 모델을 효과적으로 융합하는 성장전략이 요구되는 시점이며, 이를 실험실정보 관리 분야에 적용하면 LIMS도 새로운 국면을 맞이하게 될 것으로 기대가 되며, 해당 기술의 적용 및 융합만이 실험실정보관리시스템이 발전할 수 있는 방안이다. 대부분의 LIMS는 개별적인 기능들로 구성되어 있고, 일반적이고 표준화된 실험실정보관리 패키지나 솔루션은 전무한 상태이다. 2015년 프루스트 앤 설리번 한국지사의 에이쉬 비벡안단(Aish Vivekanandan) 연구원에 따르면 특정 요구들에 의해 별로 필요하지 않은 기능들을 지닌 제품보다 유연성을 지닌 소프트웨어를 제공해야 한다고 하며, 실험실 통합과 더 용이해진 데이터 접근성과 사용자 분석이 가능한 시스템으로의 구축이 필요하다고 한다[5].

본 논문에서는 데이터 처리의 유연성을 확보하고자 비정형 데이터를 처리하기 위한 빅데이터 수집, 처리·저장 기술을 활용하고 보다 더 많은 실험실 환경에서 활용하기

위하여 오픈 소스 기반으로 시스템을 구축하고자 한다.

2. 관련연구

2.1 실험실정보관리시스템(LIMS)

2.1.1 실험실정보관리시스템 개요

LIMS는 시험실에서 진행되는 시험 업무의 자동화 및 정보화를 통해 시험의뢰부터 최종 승인까지의 일련의 프로세스를 추적 가능하게 하며, 분석기기와 연동하여 실험실에서 발생하는 데이터의 관리도 가능하게 한다. 또한 LIMS는 사용자 권한에 따라 시험 업무의 접근 관리가 가능하며 전자서명(Electronic Signature) 및 데이터의 Audit Trail 기능을 제공한다. 이 외에도 전자시험일지(Electronic Lab Notebook), 시약 및 소모품 관리, 라벨 및 바코드 출력, 안정성 시험 관리 등 다양한 기능의 제공이 가능하다[6].

표 1. LIMS의 주요 기능
Table 1. LIMS Functionality

	Description
Audit Trail	Create, modify, delete test analysis data, and record and manage all progress with the audit function
Security	System security is maintained by appropriate system access rights and user management
Electronic Signature	Functions required by 21 CFR Part 11, such as password change cycle, automatic logout, and login failure handling as a management function for the validity of electronic signatures
Data Integrity	Integrated management of all raw data generated in the test analysis business in relation to analysis items with the integrated function of test data
Chain of custody	The traceability function provides information on samples and processing processes at any stage of data generation through LIMS
Sample Tracking	Provides a function to enable traceability of all test samples from test request to test progress and review
Stability	Provide stability test function through schedule management
Reagent	Provides management of all reagents and consumables used in the laboratory
Label & Report	Labels and barcodes can be printed through LIMS and output functions such as test reports are provided

LIMS에서의 주요기능[6]을 다음과 같은 기준으로 재분류를 하고 해당 기능을 요약하면 다음과 같다[2].

2.1.2 공통기능

LIMS를 사용하기 위하여 공통적으로 요구되는 기능으로, 파일관리 및 연계 등을 위한 기능이다. 세부적으로 기능을 분류하면, 파일업로드 및 파일 다운로드, 엑셀파일 기반의 파일 업로드 및 다운로드, 프린터 및 라벨 출력, Open API를 이용한 우편번호 찾기 기능, Open API 데이터 파싱 기능(XML, JSON 데이터 처리), 파일 첨부 및 메일보내기 등의 기능이 제공된다. 또한 사용자관리 및 권한관리 등의 LIMS 사용환경을 통합적으로 관리할 수 있는 기능도 제공한다.

2.1.3 시험성적 관리기능

실험검사를 위한 시험항목관리부터 시험완료 후 성적서 발급 등의 기능이며, 시험항목 입력, 수정 및 삭제와 검색 기능, 생성된 성적서 또는 보고서의 라벨 출력 및 프린트 출력 기능을 제공한다.

2.1.4 시험장비 및 자원 관리기능

시험장비 및 시약 등의 자원 관리를 위한 기능이며, 시험장비 등록, 수정 및 삭제 기능, 시약/초지별 재고 관리 및 엑셀 출력 기능 등이 제공된다.

2.1.5 실험검사 관리기능

제품규격관리부터 검사관리까지의 기능이며, 분류, 검체 분류, 제품 유형별 시험항목 추가, 수정 및 삭제 기능, 검사접수를 위한 검사등록 및 검색기능, 엑셀업로드를 이용한 검사요청에 대한 일괄/개별 등록 기능, 검사 완료된 시험의뢰의 히스토리 관리, 프린트 출력 여부 및 메일 발송 여부 등을 설정할 수 있는 관리기능 등이 제공된다.

2.2 빅데이터 처리 기술

2.2.1 빅데이터 기술 개요

빅데이터는 4차 산업혁명의 핵심기술 중 하나로, ICBM(IoT; Internet of Things, Cloud, BigData,

Mobile, Machine Learning 등)을 의미한다[4]. 빅데이터 기술은 수집, 저장·처리, 분석·예측, 시각화 및 응용·서비스 등의 상세기술로 분류할 수 있으며, 각각의 영역의 기능과 세부기술을 요약하면 다음과 같다[7].

2.1.2 데이터 수집기능

빅데이터 수집은 조직 내부와 외부의 분산된 여러 데이터 소스로부터 필요로 하는 데이터를 검색하여 수동 또는 자동으로 수집하는 과정과 관련된 기술로 단순 데이터 확보가 아닌 검색·수집·변환을 통해 정제된 데이터를 확보하는 기술이다. ETL, 크롤러 엔진, 로그 수집기, 센싱, RSS 및 Open API 등의 기술이 해당된다.

2.1.3 데이터 저장·처리기능

작은 데이터라도 모두 저장하여 실시간으로 저렴하게 데이터를 처리하고, 처리된 데이터를 더 빠르고 쉽게 분석하도록 하여 이를 비즈니스 의사결정에 바로 이용하는 기술과 엄청난 양의 데이터를 관리·유통·분석하는 일련의 기술이다. 병렬 DBMS, 하둡(Hadoop), NoSQL과 실시간 처리, 분산 병렬처리, 인-메모리 및 인-데이터베이스 처리 기술 등이 해당된다.

2.1.4 데이터 분석·예측기능

데이터를 효율적으로 정확하게 분석하여 비즈니스 등의 영역에 적용하기 위한 기술로 이미 여러 영역에서 활용해온 기술이다. 통계분석, 데이터 마이닝, 텍스트 마이닝, 예측 분석, 최적화, 평판분석, 소셜 네트워크 분석 등의 기술이 해당된다.

2.1.5 데이터 시각화(표현) 및 응용·서비스

자료를 시각적으로 묘사하는 기술로 빅데이터는 기존의 단순 선형적 구조의 방식으로 표현하기 힘들기 때문에 빅데이터 시각화 기술이 필수적이다. 편집기술, 정보 시각화 기술, 시각화 도구 등의 기술이 해당된다.

3. 빅데이터 처리를 위한 LIMS 설계

3.1 오픈 소스 기반의 빅데이터 수집 기술 설계

빅데이터 수집 기술은 로그 수집기, 크롤링, 센싱 등이 있으며, 해당 기술을 상세하게 나열하면 다음과 같다.

표 2. 빅데이터 수집 관련 기술
Table 1. Big Data Acquisition Technique

	Description
Log collector	조직 내부에 존재하는 웹 서버의 로그 수집 웹 로그, 트랜잭션 로그, 클릭 로그, DB 로그 데이터 등을 수집
Crawling	주로 웹 로봇을 이용하여 조직 외부에 존재하는 소셜 데이터 및 인터넷에 공개되어 있는 자료 수집
Sensing RSS Reader	각종 센서를 통해 데이터를 수집 데이터의 생산, 공유, 참여 환경인 웹 2.0을 구현하는 기술
Sqoop	Hadoop과 관계형 데이터베이스 간의 데이터 전송을 지원하는 기술로 MySQL 같은 데이터베이스로부터 하둠 분산 파일 시스템으로 데이터를 전송하는데 사용
Flume	분산환경에서 대량의 로그 데이터의 효과적으로 수집해 다른 곳으로 전송하는 서비스로 실시간 로그 분석이 가능하도록 함
Chukwa	분산 서버로부터 로그 데이터를 수집하여 하둠 클러스터의 로그나 서버의 상태정보를 관리해 하둠 파일 시스템에 저장하며 실시간 분석이 가능하도록 함
Splunk	업무 현장이나 클라우드 상에 존재하는 페타바이트급의 기록 데이터와 실시간 기계 데이터를 모니터링하고 분석함
Scribe	페이스북이 개발해 공개한 로그수집 기술로 대량의 서버에서 실시간으로 오는 로그 데이터를 집약해 하둠 분산 시스템에 로그를 저장
Kafka	LinkedIn에서 최초로 만들어졌으며, 로그 데이터를 수집할 뿐만 아니라 메시지 시스템을 통해 전송 데이터를 압축하고 메시지를 일괄적으로 전송

본 논문에서 제안하는 LIMS는 기존의 LIMS와 다르게 오픈 소스 기반의 빅데이터 수집 기술을 활용하는 플랫폼으로 재구성하였다. 이를 위하여 수집시 외부 데이터를 가져오기 위한 크롤링 기술을 적용하였다. 데이터 수집 성능과 중복되는 데이터 및 데이터 생성주기 등을 고려하여 캐시 및 메모리 기반의 수집 데이터 중복 확인 기능을 구현하고자 소프트웨어 스택을 구성(아키텍처)하였다.

오픈 소스 기반의 빅데이터 수집기술을 적용하기 위하여 Data Mart 즉, 타겟 도메인내에 속하는 다양한 기관의 웹사이트 데이터와 참조가 되는 해외 사이트에 구성된 데이터를 Raw Data 형태로 수집하여 적재하고, 성능을 향상시키기 위하여 티어별 다양한 데이터를 분류하고, 정제하여 통계 데이터로도 활용이 가능하도록 설계하였다.



그림 1. 오픈 소스 기반의 빅데이터 수집기술 적용
Fig. 1. Open source based on Big data acquisition

3.2 오픈 소스 기반의 빅데이터 저장 기술 설계

빅데이터 저장 기술은 작은 데이터라도 모두 저장하여 실시간으로 데이터를 저렴하게 처리하고, 처리된 데이터를 좀 더 빠르고 쉽게 분석하여 의사결정에 적용함에 목적이 있다. 이 때 작은 데이터를 저장하는 기술은 구글이나 애플, 야후 등에 의해 요소기술로서 상당히 높은 완성도에 도달해 있으며, 최근에는 Hadoop의 HDFS/HBase, Cassandra, MongoDB 등이 대표적인 기술이다. 한국은 ETRI의 GloryFS 등과 같은 분산파일 시스템이 존재하며, 병렬 DBMS와 NoSQL은 모두 대용량의 데이터를 저장하기 위해 수평 확장 접근 방식을 취하고 있다는 점에서 Hadoop과 유사하다고 할 수 있다.

빅데이터 처리 기술은 빅데이터에서 유용한 정보 및 숨어있는 지식을 찾아내기 위한 데이터 가공 및 분석 과정을 지원하는 기술이다. 대규모 데이터 처리를 위해 확장성, 데이터 생성 및 처리속도를 해결하기 위한 처리시간 단축 및 실시간 처리 지원, 비정형 데이터 처리 지원 기술 등이 필요하다. 대표적으로 Hadoop은 분산파일 시스템인 HDFS와 MapReduce(Hadoop 2이후 YARN)로 구현된 빅데이터 처리 기술이 있다.

본 논문에서 제안하는 LIMS는 오픈 소스 기반의 빅 데이터 저장 기술을 활용하는 플랫폼을 제안하였고, 이를 위하여 저장시 Raw Data(원천데이터)의 비정형 데이터를 저장하기 위한 NoSQL과 저장된 데이터의 검색 및 통계 등의 질의 속도 향상을 위하여 RDBMS(관계형 데이터베이스)인 MariaDB를 적용하였다. MariaDB는 MySQL과 동일한 아키텍처를 제공하며 Oracle의 유료 라이선스가 아닌 무료 라이선스를 사용하기 때문에 비용측면에서도 이득이 있어 적용하였다.

오픈 소스 기반의 빅데이터 저장기술을 적용하기 위하여 데이터를 단계별로 구성하였고, Data Lake를 구성하여 Raw Data, Meta Data, Master Data 등의 관리를 위하여 In-memory DB, NoSQL, RDBMS를 단계적으로 구성하여 Big Data 기반의 데이터 플랫폼을 구성하였다.

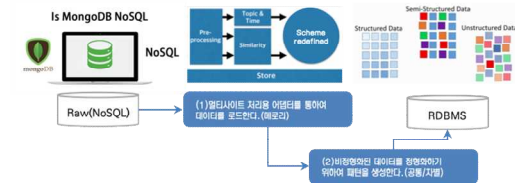


그림 2. 오픈 소스 기반의 빅데이터 저장기술 적용
Fig. 2. Open source based on Big data store

어댑터는 비정형화된 데이터의 구조를 유연하게 적용할 수 있는 Virtual Data Structure 정의 파일이며, 이를 기반으로 수집된 레퍼런스 데이터를 JSON 형태로 변환 및 key:value 구조로 패턴화하여 RDBMS에 저장할 수 있도록 구현하였다.

4. 빅데이터 처리를 위한 LIMS 구현

4.1 빅데이터 수집 기술 구현

실험실정보관리시스템(LIMS)은 레퍼런스 데이터를 수집하기 위하여 통합 LIMS외에 다양한 유사 사이트나 매체들을 이용할 수 있으며, 크롤링(Crawling, Scraping)기술을 활용하였다.

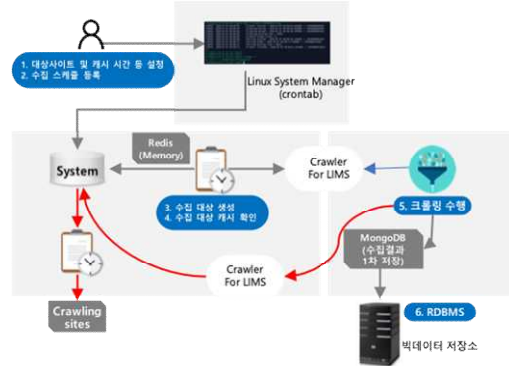


그림 3. 빅데이터 수집 : 크롤러 for LIMS
Fig. 3. Crawling for LIMS

구현한 빅데이터 수집 기술은 Python으로 구현하였으며, 모듈화(컴포넌트 또는 패키지, 유사한 개념으로 이해)하여 구성하였다.

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-
from env_logger import get_logger
logger = get_logger("crawler") # __name__
# 크롤링 대상 URL 리스트
TARGET_URLS = [
    'http://ftp.kciil.re.kr/',
    'https://www.kciil.re.kr/oaic/main.asp', # https://www.kciil.re.kr/oaic/
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=01&page=1',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=01&page=2',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=02&page=1',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=02&page=2',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=03&page=1',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=03&page=2',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=04&page=1',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=04&page=2',
    'https://www.kciil.re.kr/oaic/Document/POP/POP_EXAMIN_1IST.asp?Type=04&page=2',
    'http://cis.kciil.re.kr/oaic/main.asp', # https://www.kciil.re.kr/oaic/
    'http://cis.kciil.re.kr/fomiaSearch/1ist.do', # 구조식감사
    'http://www.chemfinder.com/',
    'http://pubchem.ncbi.nlm.nih.gov/',
    'https://chem.nlm.nih.gov/chemdplus/',
    'http://info.kciil.re.kr/oaic/', # 특허/상표정보 검색 :: 오류
    'https://cis.kciil.re.kr/regulationSearch/main.do', # 규제정보 검색
    'http://cis.kciil.re.kr/regulationSearch/1ist.do',
    'http://cis.kciil.re.kr/assessmentSearch/main.do', # 안전성정보 검색
    'http://cis.kciil.re.kr/assessmentSearch/1ist.do',
    'http://www.alltoos.biz/',
    'http://lape.kciil.re.kr/', # 특허/상표정보 검색
    'http://lape.kciil.re.kr/cotalsearch',
    'http://lape.kciil.re.kr/patent',
    'http://lape.kciil.re.kr/codemark',
    'http://lape.kciil.re.kr/rtial',
    'http://lape.kciil.re.kr/trialheredemark'
]
```

그림 4. 대상 사이트 관리 모듈
Fig. 4. target.py : Managed target sites

먼저 대상 사이트를 관리하기 위한 target.py를 구성하여, 서버내에 cronjob에 등록하여 매일 00시에 수집이 이루어지도록 하였다. 크롤러의 로깅을 위한 logger.py를 구성하여 원하는 형태(warning, error, 등)로 로그 출력이 가능하도록 하였고, 캐시 및 저장을 위한 lredis.py, mongodb.py, mysql.py 등을 모듈화하여 수집 및 저장 모듈에서는 해당 모듈을 import 하면 가능하도록 구현하였다.

또한 메인 모듈 crawler.py에서의 복잡도를 낮추어

수집과 저장을 분리하고자 collector.py 모듈을 구현하여 단독으로 수집이 가능하도록 구현하였다.

```
# 데이터 수집하기
_CRAWLED_RESULT = namedtuple('crawled', 'site_key html')
def crawling_site(url, timeout=30):
    c_site = init_key(url)
    file_name = c_site.file_name
    site_key = c_site.site_key

    # 요청 간격을 랜덤하게 선택
    sleep_time = random.choice(_RANDOM_SLEEP_TIMES)

    # 내장방기 시작을 로그에 출력
    # logger.debug("Crawling start")
    # sleep: (time) {file_name}.format(time=sleep_time, file_name=file_name)
    # 요청 대기
    time.sleep(sleep_time)

    # 사이트 유효성 검사 (verify=False : SSLError)
    r = requests.get(url, timeout=timeout, verify=False)
    try:
        html = bs(r.text, 'html') # 'html_parser'
        # logger.debug("BeautifulSoup html")
        # html = bs(r.text, 'html')
    except:
        pass

    # 크롤링 종료 로그에 출력
    logger.debug("Crawling finished")
    # file_name = file_name
    return _TARGET_T(file_name=file_name, file_content=r.content, site_key=site_key, text=r.text, html=html)
```

그림 5. 수집 모듈
Fig. 5. collector.py : Managed target sites

구현된 수집 기술의 성능을 검증하기 위하여, (1)데이터 수집을 위한 크롤러를 구동하여 (2)target에 설정된 5개의 참조 사이트를 대상으로 크롤링을 수행하도록 하였다. (3)배치 형태로 크롤러가 동작하며, (4)시험 방법 및 내용을 수집하는데 있어서 대상 사이트가 부하가 없도록 5초 이하를 목표로 구현하였다.

표 3. 크롤링 데이터 수집 시간(성능검증)
Table 1. Crawl Data Aquisition Time

회차	1	2	3	4	5	평균
수집시간	3.954	3.346	3.322	2.984	3.824	3.486

4.2 빅데이터 저장 기술 구현

앞서 언급한 것과 같이 메인 모듈의 복잡도를 낮추어 수집과 저장을 분리하였고 processor.py 모듈을 구현하여 저장 및 처리가 가능하도록 구현하였다. 또한 수집과 저장이 분리되면서 각 단위별로 오류 및 중복 데이터 등의 처리를 위하여 수집시 원천데이터를 파일(i.e.[요청페이지/시간 등으로 표현된 랜덤생성 파일명].html)로 생성하여 저장하고, processor.py에서는 파일 기반으로도 처리·저장이 가능하도록 구현하였다.

```
# [Step 4] 동일한 속성을 기준으로 address에 생성된 테이블 스키마(DDL)를 생성한다.
mysqlClient = mysql.connector.MySQLClient()
try:
    # cursor 생성
    cursor = mysqlClient.cursor()

    # Tables for create, alter, drop
    sql = "CREATE TABLE IF NOT EXISTS/ 'liass_test' (" \
        + "test_id int NOT NULL AUTO INCREMENT, " \
        + "title varchar(255) NOT NULL, " \
        + "register_date datetime NOT NULL DEFAULT CURRENT_TIMESTAMP, " \
        + "PRIMARY KEY ('test_id') " \
        + ") ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;"
    result = cursor.execute(sql)
    logger.info("Create tables")
    mysqlClient.commit()

    sql = "DROP TABLE IF EXISTS/ 'liass_test'; "
    result = cursor.execute(sql)
    logger.info("Drop tables")
    mysqlClient.commit()
except Exception as e:
    error = str(e)
    if (error.find("already exists") > 0):
        # Tables alter
        sql = "ALTER TABLE 'liass_test' ADD COLUMN 'content' VARCHAR(255) NULL AFTER 'title'; "
        result = cursor.execute(sql)
        logger.info("Alter tables")
        mysqlClient.commit()
    except Exception as e:
        error = str(e)
        if (error.find("Duplicate") > 0):
            pass
        elif (error.find("Unknown table") > 0):
            logger.info("Table is not exists")
        else:
            logger.error(e)
finally:
    mysqlClient.disconnect()
```

그림 6. 저장 및 처리 모듈
Fig. 6. processor.py : Stored and processing data

저장 및 처리 모듈의 결과는 NoSQL과 MariaDB에서 확인이 가능하지만, LIMS내의 대시보드에서도 확인이 가능하도록 구현하였다. 저장 및 처리 결과에 대한 건수(요청건수 및 완료건수, 실패건수 등)를 단순 통계로 보여주며, 크롤러가 동작하여 수집단계에서의 시간, 수집 후 처리 및 저장단계에서의 시간 등을 확인할 수 있다.

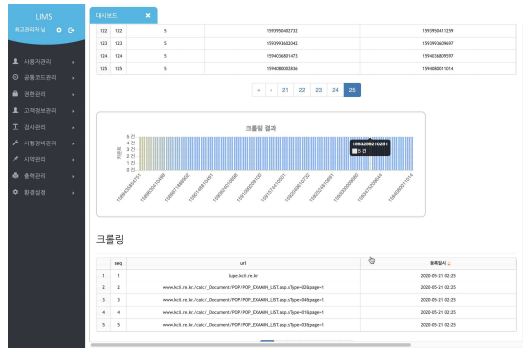


그림 7. 실험실정보관리시스템 대시보드
Fig. 7. Dashboard on LIMS web site

구현된 저장 기술 성능을 검증하기 위하여, (1)수집된 데이터를 처리하여 데이터베이스에 저장하는 시간을 측정하였고, (2)Raw data 형태(html)로 NoSQL 저장하는 1단계와 (3)정제된 데이터(시험 방법 및 내용)를 RDBMS에 저장하는 2단계, (4)수집 및 처리 완료된 성공/실패에 대한 사이트별 결과 통계 데이터를 저장하는 3단계로 구성하였다. 저장 기술 성능은 1~3

단계까지의 결과 저장 시간을 20초 이하를 목표로 구현하였다.

표 4. 크롤링 데이터 처리 시간(성능검증)
Table 4. Crawl Data Stored Time

회차	1	2	3	4	5	평균
처리시간	4.630	4.839	4.565	4.414	4.877	4.656

저장된 레퍼런스 데이터는 시험 방법 및 내용을 실험자가 참고할 수 있도록 검색 기능을 제공하고 검색된 시험 방법 및 내용을 참고하여 시험의 진행 및 결과를 기록할 수 있도록 구현하였다.

5. 결론

본 논문에서는 오픈 소스 기반의 빅데이터 수집 및 저장 기술을 활용한 LIMS 고도화 제안 및 시스템 구현을 목표로 진행하였다. 먼저 기존 적용되어 사용되던 LIMS의 기능 등을 분석하여 빅데이터 기술 활용분야를 도출하였고, 이를 적용하고자 빅데이터 기술의 전반적인 분석과 특히 필요하다고 도출된 수집 및 저장 분야에 적용한 LIMS 설계 및 시스템을 구현하는 단계로 진행하였다. 관련연구를 통하여 LIMS는 정유, 석유화학, 정밀화학, 제조업, 금속, 제철, 식품, 의약, 연구소, 정밀화학, 보건환경, 검사소 등 여러 분야의 실험실내 시험 분석 업무에는 어느 곳이든 적용 가능하며, 다양한 분야에서 활용이 되는 만큼 시험 업무를 수행함에 있어 시험 내용과 방법에 대한 참조가 필요하다, 이를 위하여 오픈 소스 기반의 빅데이터 수집기술 및 전처리, 저장, 관리 등의 기술 활용이 가능하다는 점이 강점이라 할 수 있다. 강점을 살리면서 LIMS의 확산 및 확대를 위하여 시스템의 설계 및 구현은 빅데이터 기술의 사용사례로서 다양한 업무분야에 빅데이터 기술이 적용 가능하고 필요하다는 당위성을 대변할 수 있을 것으로 기대한다. 향후에는 실험실에서 진행되는 시험뿐만 아니라 결과의 히스토리 관리를 통하여 기존 시스템의 외부적인 데이터 활용뿐만 아니라 지속적으로 증가하는 내부 데이터의 연계를 통한 시스템 고도화 및 안정화, 또한 운영으로서의 실험실정보관리시스템(DevOps for LIMS)에 대한 연구 및 시스템 설계 및 구현이 필요한 것으로 사료된다.

REFERENCES

- [1] Yullin Technologies Co., Ltd., "LIMS/LAS Overview", <http://www.yullin.com/Lims>.
- [2] Wiseman Lim, Ryu Ki Sang, "Development of Cloud-based Laboratory Information Managemnet System using Bigdata Acquisition, Processing·Store Technology", Regional Enterprise Open-Innovative Voucher Program(R&D, P0010684) 2019.
- [3] Autoscribe Informatics, "LIMS Overview", <https://www.autoscribeinformatics.com/lims-laboratory-information-management-system>.
- [4] IITP(Institute of Information & Communications Technology Planning & Evaluation, "About the Forth Industrial Revolution", 2017.
- [5] FROST, "Analysis of the US MLIMS Market", November, 2014.
- [6] Choi Byung Lee, "A Study on the Implementation Strategy for Laboratory Information Management System of the Pharmaceutical Company", Industiral Pharmaceutical Sciences, The Graduate School, Yonsei University, pp. 4, June, 2020.
- [7] Lee Jae Jun, "Hallum ICT Policy Journal", CPRC ISSUE BRIEFING HALLYM, No.2, pp. 14-19, December, 2015.

저자약력

김 순 권(Soon-Gohn Kim)

[중신회원]



- 1999년 : 전북대학교 일반대학원 전자계산기공학과(공학박사)
- 1987년: 동국대학교 교육대학원 전산교육학과(교육학석사)
- 1979년 : 전북대학교 공과대학 자원공학과(공학사)
- 1982년~1987년: 동아생명보험(주) 전자계산실 근무
- 1987년~1995년: 한국원자력연구소 선임연구원
- 1995년~현 재 : 중부대학교 소프트웨어공학부 교수

<관심분야> 데이터베이스, 정보보호, 유비쿼터스컴퓨팅, 모바일 컴퓨팅, 빅데이터컴퓨팅, 정보시스템관리 등