Journal of the Korea Convergence Society
Vol. 12. No. 7, pp. 45-51, 2021

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2021.12.7.045

빅데이터 수집 처리를 위한 분산 하둡 풀스택 플랫폼의 설계

이명호
세명대학교 정보통신학부 교수

Design of Distributed Hadoop Full Stack Platform for Big Data Collection and Processing

Myeong-Ho Lee
Professor, School of Information Communication, Semyung University

요 약 급속한 비대면 환경과 모바일 우선 전략에 따라 해마다 많은 정형/비정형 데이터의 폭발적인 증가와 생성은 모든 분야에서 빅데이터를 활용한 새로운 의사 결정과 서비스를 요구하고 있다. 그러나 매년 급속히 증가하는 빅데이터를 활용하여 실무 환경에서 적용 가능한 표준 플랫폼으로 빅데이터를 수집하여 적재한 후, 정재한 빅데이터를 관계형 데이터베이스에 저장하고 처리하는 하둡 에코시스템 활용의 참조 사례들은 거의 없었다. 따라서 본 연구에서는 스프링 프레임워크 환경에서 3대의 가상 머신 서버를 통하여 하둡 2.0을 기반으로 쇼셜 네트워크 서비스에서 키워드로 검색한 비정형 데이터를 수집한 후, 수집된 비정형 데이터를 하둡 분산 파일 시스템과 HBase에 적재하고, 적재된 비정형 데이터를 기반으로 형태소 분석기를 이용하여 정형화된 빅데이터를 관계형 데이터베이스에 저장할 수 있게 설계하고 구현하였다. 향후에는 데이터 심화 분석을 위한 하이브나 머하웃을 이용하여 머신 러닝을 이용한 클러스터링과 분류 및 분석 작업 연구가 지속되어야 할 것이다.

주제어 : 모바일 우선 전략, 빅데이터, 하둡 에코시스템, 스프링 프레임워크, 하둡 분산 파일 시스템, 형태소 분석기

Abstract In accordance with the rapid non-face-to-face environment and mobile first strategy, the explosive increase and creation of many structured/unstructured data every year demands new decision making and services using big data in all fields. However, there have been few reference cases of using the Hadoop Ecosystem, which uses the rapidly increasing big data every year to collect and load big data into a standard platform that can be applied in a practical environment, and then store and process well-established big data in a relational database. Therefore, in this study, after collecting unstructured data searched by keywords from social network services based on Hadoop 2.0 through three virtual machine servers in the Spring Framework environment, the collected unstructured data is loaded into Hadoop Distributed File System and HBase based on the loaded unstructured data, it was designed and implemented to store standardized big data in a relational database using a morpheme analyzer. In the future, research on clustering and classification and analysis using machine learning using Hive or Mahout for deep data analysis should be continued.

Key Words : Mobile Frist Strategy, Big Data, Hadoop Ecosystem, Spring Framework, HDFS, Morpheme Analyzer

*This study has been supported by Semyung University Research Year 2020.
*Corresponding Author : Myeong-Ho Lee(mhlee@semyung.ac.kr)
Received April 10, 2021 Revised May 18, 2021
Accepted July 20, 2021 Published July 28, 2021

1. 서론

IDC의 Global DataSphere에 따르면 2020년 전 세계에서 59ZB 이상의 데이터가 생성, 캡처, 복사 및 소비할 것으로 예상하며, COVID-19은 재택 근무자의 업무 수를 갑작스럽게 증가시키고 생성되는 데이터의 혼합을 화상 통신 및 다운로드된 데이터 소비와 스트리밍 비디오의 가시적인 증가를 포함하는 더 풍부한 데이터 세트로 변경함으로써 이 수치에 기여하고 있다고 분석하였다[1]. 특히 Domo의 "Data Never Sleeps" 인포그래픽에 따르면 2020년에 COVID-19 유행으로 인해 더 많은 사람들이 실내에서 그리고 웹 기반 앱과 도구로 서로 연결되어 더 넓은 세상을 유지하도록 강요했기 때문에 디지털 세계의 데이터 소비는 더욱 강회될 것으로 예상하였다[2].

2019년 대비 2020년 Internet Minute의 데이터 소비 트렌드 증가율을 중요한 소비 주체별로 보면 Table 1과 같다[3,4].

Table 1. Internet Minute Comparison by Year

Items	Units	2019	2020	Rate (%)
YouTube	Videos Viewed	4,500,000	4,700,000	4
NETFLIX	Hours Watched	694,444	764,000	10
Google	Search Queries	3,800,000	4,100,000	8
facebook	Loggin in	1,000,000	1,300,000	30
Instagram	Scrolling Instagram	347,222	694,444	100
Twitter	People Tweeting	87,500	194,444	122
Email	Emails Sent	188,000,000	190,000,000	1
Online Shopping	Spent Online($)	996,956	1,100,000	10

이와 같이 데이터의 급격한 증가와 소비는 향후 더욱 가속화될 예정이며, 이러한 비정형 이종의 빅데이터 안에 숨어 있는 다양한 패턴들을 수집/정제하고 분석하여 미래의 성장동력을 찾는 것이 기업들의 새로운 비즈니스 모델이 되고 있다.

이러한 대용량 빅데이터 처리를 여러대의 컴퓨터에서 분산 처리할 수 있게 해주는 프레임워크가 하둡(Hadoop)이다. 하둡은 아파치 루씬의 창시자인 더그 커팅이 2003년 구글의 분산 파일 시스템 아키텍처 논문과 2004년 맵리듀스 논문을 기반으로 HDFS과 MapReduce를 개발과 적용에서 시작되었다[5,6]. 2011년 발표된 하둡 1.0은 맵리듀스를 실행할 때 맵리듀스 작업 갯수를 관리했기 때문에 클러스터 전체 사용률이 낮은 단점이 있었다. 2012년 잡트랙커의 병목현상을 제거하기 위하여 슬롯이 아닌 컨테이너 단위로 리소스를 할당하는 YARN이 도입된 것이 하둡 2.0이다. 하둡 2.0은 맵과 리듀스의 관계가 1:1 관계가 아니며 하나 이상의 컨테이너를 실행할 수 있기 때문에 잡들의 워크로드를 고려하여 리소스 설정을 진행할 수 있는 장점이 있다. 2017년 하둡 3.0은 erasure coding, YARN timeline service v.2 등이 도입되었다[7,8].

차세대 웹 표준를 대비하여 우리나라에서도 2008년도부터 오픈소스를 적극 활용한 전자정부 표준프레임워크를 구성하였다. 2021년 현재 스프링 프레임워크 5.2.5와 스프링 부트 2.2.x를 지원하는 전자정부 표준프레임워크 실행환경 4.0.0(alpha) 업데이트를 발표하였다[9].

그러나 현재까지 대부분의 연구는 하둡 기반 딥 러닝 프레임워크 기술 동향 연구[10]나 웹 서비스 데이터 처리 설계 및 구현 연구[11] 그리고 실시간 데이터가 아닌 샘플 데이터를 통하여 수집과 정제를 가정한 후 처리와 탐색, 분석을 위한 연구였다. 또한 하둡 기반의 빅데이터를 활용하는 분야는 전 영역에 걸쳐 점점 늘어나고 있지만 엔터프라이즈 분산 기반의 스프링 프레임워크 환경에서 실시간으로 데이터를 수집한 후 의미있는 데이터로 정제해야 하는 연구는 미비하였다. 또한 하둡 에코시스템과 스프링 프레임워크를 기반으로 실시간 스크래핑을 통하여 비정형 빅데이터를 수집한 후 형태소 분석기를 통하여 비정형 빅데이터를 정제하고 의미있는 정형화된 빅데이터로 변환하여 관계형 데이터 베이스로 저장 처리하는 분산 플랫폼 설계 연구는 많이 미비하였다.

따라서 본 연구에서는 스프링 프레임워크 환경에서 3대의 가상 머신 서버을 통하여 하둡 2.0을 기반으로 SNS 상에서 키워드로 검색한 비정형 빅데이터를 수집한 후, 수집된 빅데이터를 HBase에 적재한다. 적재된 비정형 빅데이터를 기반으로 형태소 분석기를 이용하여 키워드, 제목, 내용 등의 정형화된 빅데이터를 오라클 데이터베이스에 저장할 수 있도록 하였다. 이러한 지속적인 빅데이터를 수집과 저장을 통하여 필요한 빅데이터 활용 풀스택 플랫폼 시스템을 설계하고 참조 모델을 구현하였다.

2. 관련 연구에 대한 고찰

2.1 수집 플랫폼

다양한 다수의 서버로 부터 빅데이터를 수집하는 다양한 저장 플랫폼이다. 그 중에서도 플룸(Flume)은 아파치

오픈소스 프로젝트로 대용량의 로그를 수집할 수 있도록 여러가지 기능을 제공하는 프로그램이다[12]. 카프카(Kafka)도 아파치 오픈소스로 데이터 파이프라인을 구축할 때 많이 고려되는 분산 메시징 시스템 중의 하나로 대용량의 실시간 로그 처리에 특화된 설계를 기반으로 기존 시스템 보다 우수한 성능을 자랑한다[13]. 스톰(Storm)은 아파치 오픈소스로 분산 스트리밍 프로세싱 연산 프레임워크이다[14]. 에스퍼(Esper)는 실시간으로 발생하는 복잡한 이벤트 처리(CEP) 및 이벤트 스트림 처리(ESP)를 위한 오픈 소스 자바 기반 소프트웨어이다[15].

2.2 저장 플랫폼

수집된 실시간 비정형 데이터를 적재 및 저장하는 하둡 에코시스템의 플랫폼으로는 데이터를 분산 환경에서 분리하여 저장, 처리, 요청할 수 있도록 구성된 파일 시스템인 HDFS 위에서 만들어진 분산 컬럼 기반의 데이터베이스인 HBase가 있다. HBase의 마스터 서버는 테이블을 위한 분산 및 가용성의 기본 단위인 리전(Region)을 리전 서버에 할당하고 할당 업무를 위한 주키퍼의 도움을 받는 작업을 수행한다[16]. Redis는 모든 데이터를 메모리로 불러와서 처리하는 메모리 기반의 key-value 구조이다. 대용량 데이터를 메모리 상에 저장 및 조회함으로써 빠르게 읽기와 쓰기의 속도를 보장하는 비 관계형 데이터베이스이다[17].

2.3 처리/탐색 플랫폼

초기의 하둡 기반 플랫폼은 배치 처리의 강점을 가지고 있었으나 실시간 처리 등의 한계로 인하여 추가적인 에코 시스템들이 포함되고 지속적인 기능 개선이 일어났다. 그 중에서도 인메모리 기반의 대용량 데이터 고속 처리 엔진을 가진 범용 분산 클러스터인 아파치 오픈소스 컴퓨터 프레임워크인 스파크(Spark)가 있다[18]. 하둡에서 동작하는 데이터 웨어하우스 구조인 하이브(Hive)는 SQL 형식의 쿼리 작성으로 데이터 질의나 분석 기능이 가능하다[19]. 또한 하둡의 잡을 관리하기 위한 서버 기반의 워크플로는 이벤트 발생시 액션을 수행하는 스케쥴링, 조정 및 관리 시스템으로 아파치 우지(Oozie)와 에어플로(Airflow)가 있다[20,21]. 또한 스쿱(Sqoop)은 관계형 데이터베이스와 하둡 간의 빅 데이터들을 효율적으로 변환하여 주는 CLI 애플리케이션이다. 스쿱은 HDFS으로 데이터들을 맵리듀스로 변환한 후, 그 데이터들을 RDBMS로 내보낼 수 있다[22].

2.4 분석/응용 플랫폼

기계학습 유형에는 훈련 데이터로 부터 회귀분석과 분류를 위한 하나의 함수를 유추해 내기 위한 지도학습 분야와 행동심리학에서 영감을 받아 어떤 환경 안에서 정의된 에이전트가 현재의 상태를 인식하여 선택 가능한 행동들 중 보상을 최대화하는 행동 혹은 행동 순서를 선택하는 강화학습 분야와 비지도학습 분야가 있다[23]. 이러한 접근방법별 알고리즘을 기반으로 분산처리가 가능하고 확장성을 가진 기계학습용 라이브러리가 아파치 머하웃(Mahout)이다 [24]. 아파치 임팔라(Impala)는 대규모 병렬처리 SQL 엔진이다[25].

2.5 주키퍼

주키퍼(Zookeeper)는 분산 환경에서 하나의 서버에만 서비스가 집중되지 않도록 서버들 간의 조정으로 서비스를 분산 처리하게 해주며, 서버들 간의 환경 설정을 통합적으로 관리해 준다. 또한 실행 서버가 문제가 발생하면 다른 대기 서버가 실행 서버로 스위칭되어 서비스가 중지없이 동작하며, 서버들 간의 동기화를 통하여 데이터 안정성을 보장해 준다. 클라이언트가 리전 서버들(ResionServers)과 통신하려면 주키퍼를 통해야 한다[26].

2.6 하둡

하둡 에코 시스템에서 아파치 하둡의 영역을 보면 파일저장, 자원 관리, 맵리듀스, 스트림, 그래프, 메시지 전달 인터페이스 영역으로 분류할 수 있다. 이제 초창기의 빅데이터 개념이 하둡으로 통하게 되었다. 하둡 2.x와 하둡 3.x의 기능별로 비교해 보면 Table 2와 같다[27].

Table 2. Feature comparision of Hadoop 2.x vs 3.x

Features	Hadoop 2.x	Hadoop 3.x
Min Java Version Required	Java 7	Java 8
Fault Tolerance	Via replication	Via erasure coding
Storage Scheme	3x replication factor for data reliablility, 200% overhead	Erasure coding for data reliability, 50% overhead
Yarn Timeline Service	Scalability issues	Highly scalable and reliable
Standby NameNode	Supports only 1 SBNN	Supports only 2 or more SBNN
Heap Management	We need to configure HADOOP_HEAPSIZE	Provides auto-tuning of heap

2.7 스프링 프레임워크

스프링 프레임워크는 우리나라에서도 전자정부 표준 프레임워크로 채택하여 운영하는 프레임워크이다. 스프링 프레임워크의 핵심 요소는 애플리케이션 수준의 인프라 지원이다. 그리고 엔터프라이즈 애플리케이션팀이 애플리케이션 수준의 비즈니스 로직에만 집중할 수 있도록 한다[28]. 스프링 MVC는 디스패처서블릿, 핸들링매핑, 컨트롤러, 뷰리졸버로 각 컴포넌트들의 역할이 명확하게 분리하여 백엔드와 프론트엔드 혹은 퍼블리셔가 동시에 개발할 수 있는 용이성이 있다[29].

따라서 본 연구에서는 분산 빅데이터 풀스택 플랫폼를 위하여 전자정부 표준 프레임워크인 스프링 프레임워크를 기반으로 설계하였다.

2.8 마이바티스

마이바티스(MyBatis)는 개발자가 지정한 SQL과 저장 프로시저 및 고급 매핑을 지원하는 최고 수준의 퍼시스턴스 프레임워크이다. 마이바티스는 거의 모든 JDBC 코드와 매개변수의 수동 설정 및 결과 검색을 대신해 준다. 마이바티스는 데이터베이스 레코드에 기본 타입과 Map 인터페이스 그리고 자바 POJO를 설정해서 매핑하기 위해 XML과 애노테이션을 사용할 수 있다[30]. 본 연구에서는 스프링 프레임워크를 통하여 수집 및 처리된 빅데이터들을 하둡을 통하여 향후 분석 및 응용을 위하여 빅데이터를 영구 저장하기 위한 효율적인 데이터 매퍼로 마이바티스를 적용하였다.

2.9 웹 스크래핑

웹 스크래핑(Web Scaping)이란 웹 사이트 상에서 원하는 정보를 자동으로 추출하여 수집하는 기술이다. 웹 크롤링도 일종의 웹 스크래핑 기술이다. 웹 크롤링은 조직적으로 자동화된 방법으로 WWW를 탐색하는 컴퓨터 프로그램인 웹 크롤러가 하는 작업을 말하며 여러 인터넷 사이트의 페이지들을 브라우징하는 행위이다. 파싱이란 웹 페이지들의 자료에서 데이터의 특정 패턴들을 추출한 후, 추출된 정보를 가공하는 것이다[31]. 본 연구에서는 스프링 프레임워크 상에서 웹 스프레핑을 통하여 수집된 비정형 빅 데이터들을 형태소 분석기를 통하여 원하는 데이터들로 정형화하여 분류하도록 한다.

2.10 형태소 분석기

형태소 분석이란 형태소 보다 큰 어절이나 문장을 최소 의미 단위로 분절하는 과정이다. 오픈 소스의 형태소 분석기로는 꼬꼬마[32], 코모란[33], 한나눔[34], 은전한닢[35], Okt[36], khaiii[37] 등이 있다. 본 연구에서는 성능이 뛰어난 Okt 형태소 분석기를 사용하여 하둡 HBase에서 수집된 비정형 데이터를 형태소 분석을 하여 오라클 데이터 베이스에 정형화된 데이터를 저장하도록 설계하였다.

3. 개발 플랫폼의 설계

3.1 분산 빅데이터 풀스택 개발 환경

본 연구에서는 Fig. 1과 같이 스프링 프레임워크를 미들 티어로, 하둡 에코시스템을 빅데이터 티어로 두고, 오라클 데이터베이스를 데이터 티어로 하여 오라클 VirtualBox[38]로 우분투 리눅스 서버로 3대의 가상머신 통하여 빅데이터 하드웨어 환경을 구성하였다.

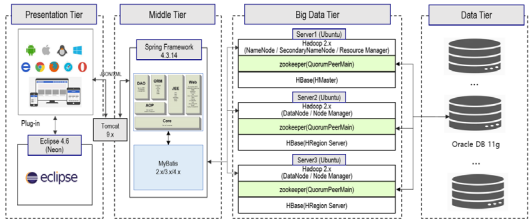

Fig. 1. Architecture of Big Data Full Stack Platform

서버1은 하둡과 HBase, 주키퍼를 설치하여 마스터 서버로 구성하였다. Fig. 2는 동작 중인 서버1의 실행 엔진들이다.

```
ubuntu@server1:~/hbase/bin$ sudo jps
2401 HMaster
1831 ResourceManager
2472 Jps
2202 QuorumPeerMain
1674 SecondaryNameNode
1470 NameNode
```

Fig. 2. Execution Environment of Server1

Fig. 3은 서버2와 서버3에서 동작 중인 실행 엔진들이다.

```
ubuntu@server2:~$ sudo jps
1446 NodeManager
1976 Jps
1353 DataNode
1834 HRegionServer
1711 QuorumPeerMain
```

```
ubuntu@server3:~$ sudo jps
1377 DataNode
1986 Jps
1699 QuorumPeerMain
1812 HRegionServer
1471 NodeManager
```

Fig. 3. Execution Environment of Server2 & Server3

따라서 본 연구에서는 Table 3과 같이 1대의 컴퓨터에서 3대의 가상 머신으로 구성하여 시스템을 구현하였다.

Table 3. Environment of Distributed Hadoop Full Stack Platform

Items	Contents
Server 1, 2, 3 O/S	Linux Ubuntu 16.04
IDE Tools	Eclipse Neon (4.6)
Web Container	Apache Tomcat 9.0.13
Java Development Kit	Linux x64 Java SE 8.x
Framework	Spring Framework 4.3.14
Hadoop Ecosystem	Hadoop 2.6.5
	HBase 1.2.6
	Zookeeper 3.4
DBMS	Oracle DataBase 11g 11.2.0

3.2 빅데이터 수집 및 처리 플랫폼 설계

빅데이터 풀스택 플랫폼 시스템은 빅데이터 수집부터 시작된다. 빅데이터 수집은 광범위하게 내부 및 외부 데이터 시스템까지 매우 다양하며, 공공데이터 포털 사이트에서는 오픈 API 형태로 JSON이나 XML과 같은 다양한 방식을 제공하고 있다[39,40]. 본 연구에서는 네이버 오픈 API에서 블로그 데이터 중에서 관심 있는 키워드 검색에 따라 실시간 빅 데이터를 수집한 후, 분산 하둡 풀스택 플랫폼 시스템으로 수집 및 처리하도록 설계하였다.

4. 분산 빅데이터 풀스텍 플랫폼의 구현

본 연구에서 구현한 분산 빅데이터 풀스택 플랫폼을 실행한 메인 화면은 Fig. 4와 같다.

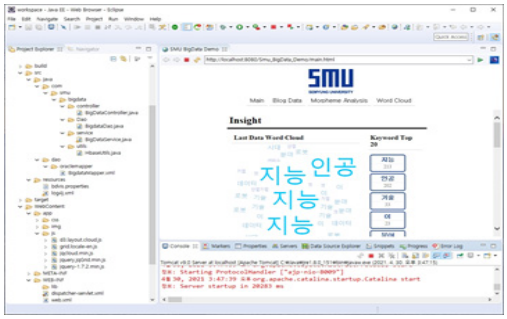

Fig. 4. Main View of BigData Full Stack Platform

데이터베이스 스키마 설계를 기반으로 비정형 데이터 저장 현황을 보면 Fig. 5와 같다.

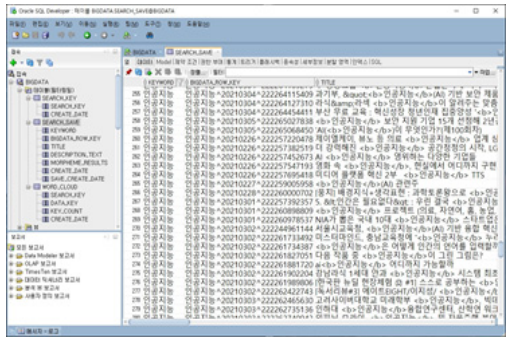

Fig. 5. DataBase Schema of BigData Full Stack Platform

서버2와 서버3에서 동작되고 있는 하둡 데이터노드의 상태를 보면 Fig. 6과 같다.

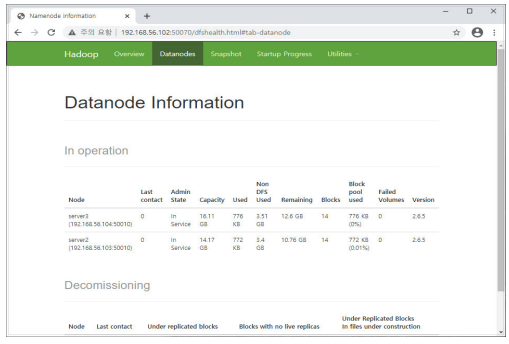

Fig. 6. Datanode Information View of Server2 & Server3

마스터 서버1에서 동작 중인 지역 서버2와 지역 서버3의 HBase 모니터링 화면을 보면 Fig. 7과 같다.

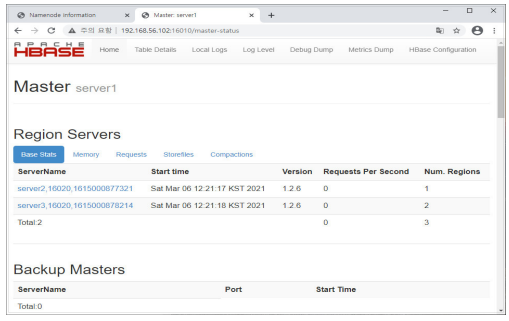

Fig. 7. HBase Master View of Region Servers

위와 같이 하둡 분산 빅데이터 풀스택 환경으로 구현

된 시스템에서 키워드를 통하여 수집된 비정형 데이터들의 결과를 보면 Fig. 8과 같다.

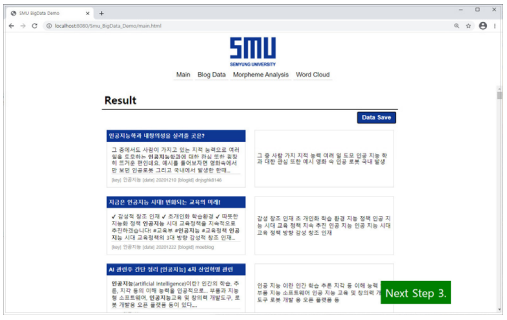

Fig. 8. Unstructured Data of Hadoop Full Stack Platform

다음은 키워드로 검색한 블로그 빅데이터를 하둡에 적재한 후 트위터 형태소 분석기를 통한 정형화된 중요 키워드 20개를 추출한 결과는 Fig. 9와 같다.

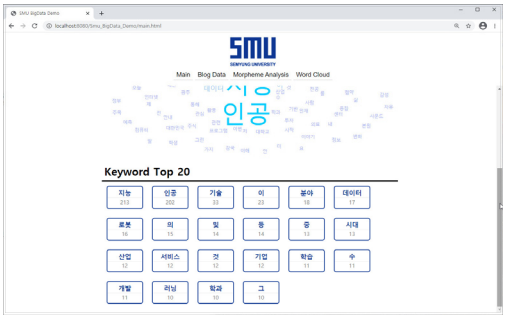

Fig. 9. Keyword Top 20 After Running a Morpheme Analyzer

이상과 같이 최종 완성된 정형화된 빅데이터들은 관계형 데이터베이스인 오라클에 저장되어 향후 지속적으로 분석 및 응용에 활용될 수 있게 하였다. 이러한 결과들을 워드클라우드를 사용한 데이터 시각화는 Fig. 10과 같다.

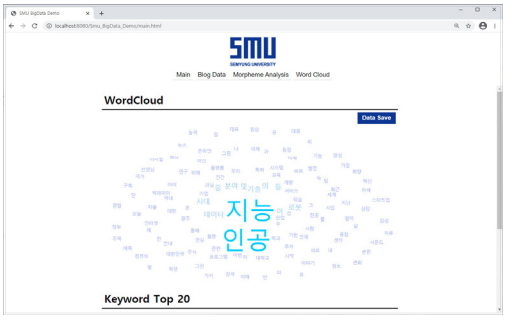

Fig. 10. WordCloud Visualization of BigData

5. 결론

앞으로 비대면 정보시스템의 환경이 다양한 디바이스의 출현과 모바일 우선 전략으로 모든 분야에서 많은 변화의 바람이 불고 있다. 또한 해마다 많은 정형/비정형 데이터의 폭발적인 증가와 생성은 모든 분야에서 빅데이터를 활용한 새로운 의사 결정과 서비스를 요구하고 있다.

그러나 매년 급속히 증가하는 빅데이터를 활용하여 서비스에 적용하는 다양한 사례들은 많지만 각 분야의 실무사례를 중심으로 소개하는 것이 대부분이었다. 또한 실무 환경에서 적용 가능한 전자정부 표준 플랫폼으로 채택된 스프링 프레임워크를 활용하여 빅데이터를 수집하고 적재한 후 정재한 빅데이터를 관계형 데이터베이스에 저장하고 처리하는 하둡 에코시스템 활용의 참조 사례들은 많이 부족하였다.

따라서 본 연구에서는 스프링 프레임워크 환경에서 3대의 가상 머신 서버를 통하여 하둡 2.0을 기반으로 쇼셜 네트워크 서비스에서 키워드로 검색한 비정형 데이터를 수집한 후, 수집된 비정형 데이터를 하둡 분산 파일 시스템과 HBase에 적재하고, 적재된 비정형 데이터를 기반으로 형태소 분석기를 이용하여 정형화된 빅데이터를 관계형 데이터베이스에 저장할 수 있게 설계하고 구현하였다. 또한 이렇게 저장된 정형화된 빅데이터를 분석하고 응용할 수 있도록 빅데이터 활용 풀스택 플랫폼을 설계하고 참조 모델을 구현하였다.

향후에는 실제 적용 사례를 통하여 데이터 심화 분석을 위한 하이브나 머하웃을 이용하여 머신 러닝을 이용한 클러스터링과 분류 및 분석 작업 연구을 위한 인공지능, 머신 러닝, 딥러닝 연구가 지속되어야 할 것이다.

REFERENCES

[1] IDC. (2020. May). Worldwide *Global DataSphere Forecast.* Global DataSphere. https://www.idc.com/getdoc.jsp?containerId=prUS46286020

[2] Domo. (2020. August). *Data Never Sleeps 8.0.* https://www.domo.com/news/press/domo-releases-eighth-annual-data-never-sleeps-infographic

[3] Lori Lewis. (2020). *This Is What Happens In An Internet Minute.* https://lorilewismedia.com/

[4] visually. (2020). *This Is What Happens In An Internet Minute.* https://visual.ly/community/Infographics/other/

what-happens-internet-minute-2020

[5 J. H. Park, S. Y. Lee, D. H. Kang & J. H. Won. (2013). Hadop and MapReduce. *Journal of Korea Data & Information Science Society, 24(5),* 1013-1027.

[6] S. B. Heo, D. C. Kang & J. Y. Choi. (2019). Hadoop based Deep Learning Framework Technology Trend. *Communications of the Korean Institute of Information Scientists and Engineers, 37(10),* 25-31.

[7] *Apache Hadoop 2.10.1*, https://hadoop.apache.org/docs/r2.10.1/

[8] *Apache Hadoop 3.1.4*, https://hadoop.apache.org/docs/r3.1.4/

[9] National Information Society Agency. *eGovFrame.* https://www.egovframe.go.kr/home/sub.do?menuNo=32

[10] S. B. Heo, D. C. Kang & J. Y. Choi. (2019). Technology Trends of Deep Learning Framework on Hadoop YARN. *Communications of the Korean Institute of Information Scientists and Engineers, 37(10),* 25-31.

[11] H. J. Kim. (2015). Design and Implementation of an Efficient Web Services Data Processing Using Hadoop-Based Big Data Processing, *Journal of the Korea Academia-Industrial cooperation Society, 16(1),* 726-734.
DOI : 10.5762/kais.2015.16.1.726

[12] Wikipedia. *Apache Flume.* https://en.wikipedia.org/wiki/Apache_Flume

[13] Wikipedia. *Apache Kafka.* https://en.wikipedia.org/wiki/Apache_Kafka

[14] Wikipedia. *Apache Storm.* https://en.wikipedia.org/wiki/Apache_Storm

[15] Wikipedia. *Esper.* https://en.wikipedia.org/wiki/Esper_(software)

[16] *Apache HBase.* https://hbase.apache.org/

[17] Wikipedia. *Redis.* https://en.wikipedia.org/wiki/Redis

[18] Wikipedia.. *Apache Spark.* https://en.wikipedia.org/wiki/Apache_Spark

[19] Wikipedia. *Apache Hive.* https://en.wikipedia.org/wiki/Apache_Hive

[20] Wikipedia. *Apache Oozie.* https://en.wikipedia.org/wiki/Apache_Oozie

[21] *Apache Airflow*, https://airflow.apache.org/

[22] Wikipedia. *Apache Sqoop.* https://en.wikipedia.org/wiki/Sqoop

[23] Wikipedia. *Machine Learning.* https://en.wikipedia.org/wiki/Machine_learning

[24] Wikipedia. *Apache Mahout.* https://en.wikipedia.org/wiki/Machine_learning

[25] Wikipedia. *Apache Impala.* https://en.wikipedia.org/wiki/Apache_Impala

[26] *Apache Zookeeper.* http://zookeeper.apache.org/

[27] *Data Flair.* https://data-flair.training/blogs/hadoop-2-vs-hadoop-3/

[28] Wikipedia. *Spring Framework.* https://en.wikipedia.org/wiki/Spring_Framework

[29] I. M. Lee. (2012). *Spring 3.1 of Toby(Vol. 1).* Seoul : Acorn.

[30] MyBatis. (2021. April). https://mybatis.org/mybatis-3/index.html

[31] Wikipedia. *Web scraping.* https://en.wikipedia.org/wiki/Web_scraping

[32] SNU IDC. *Kkma.* http://kkma.snu.ac.kr/

[33] Shineware. *Komoran.* https://www.shineware.co.kr/products/komoran/

[34] SWRC. *Han nanum.* http://semanticweb.kaist.ac.kr/hannanum/

[35] Atlassian. Bitbucket, *MeCab-ko.* https://bitbucket.org/eunjeon/mecab-ko-dic/src/master/

[36] Twitter. *Open Korea Text(Okt).* https://github.com/open-korean-text/open-korean-text

[37] Kakao Tech. *Khaiii.* https://tech.kakao.com/2018/12/13/khaiii/

[38] Wikipedia. *VirtualBox.* https://www.virtualbox.org/wiki/VirtualBox

[39] National Information Society Agency. *Publicdata Portal,* https://www.data.go.kr/

[40] Naver Developers. *Naver Open API.* https://developers.naver.com/docs/search/blog/

이 명 호(Myeong-Ho Lee) [종신회원]

- 1984년 2월 : 아주대학교 산업공학과 (공학사)
- 1986년 2월 : 아주대학교 대학원 산업공학과(공학석사)
- 2001년 2월 : 아주대학교 대학원 산업공학과(공학박사)
- 2002년 3월 ~ 현재 : 세명대학교 정보통신학부 교수
- 관심분야 : N-Tier Full Stack 프로그래밍, Spring/Spring Boot Framework, Full Stack 개발, Infographics
- E-Mail : mhlee@semyung.ac.kr