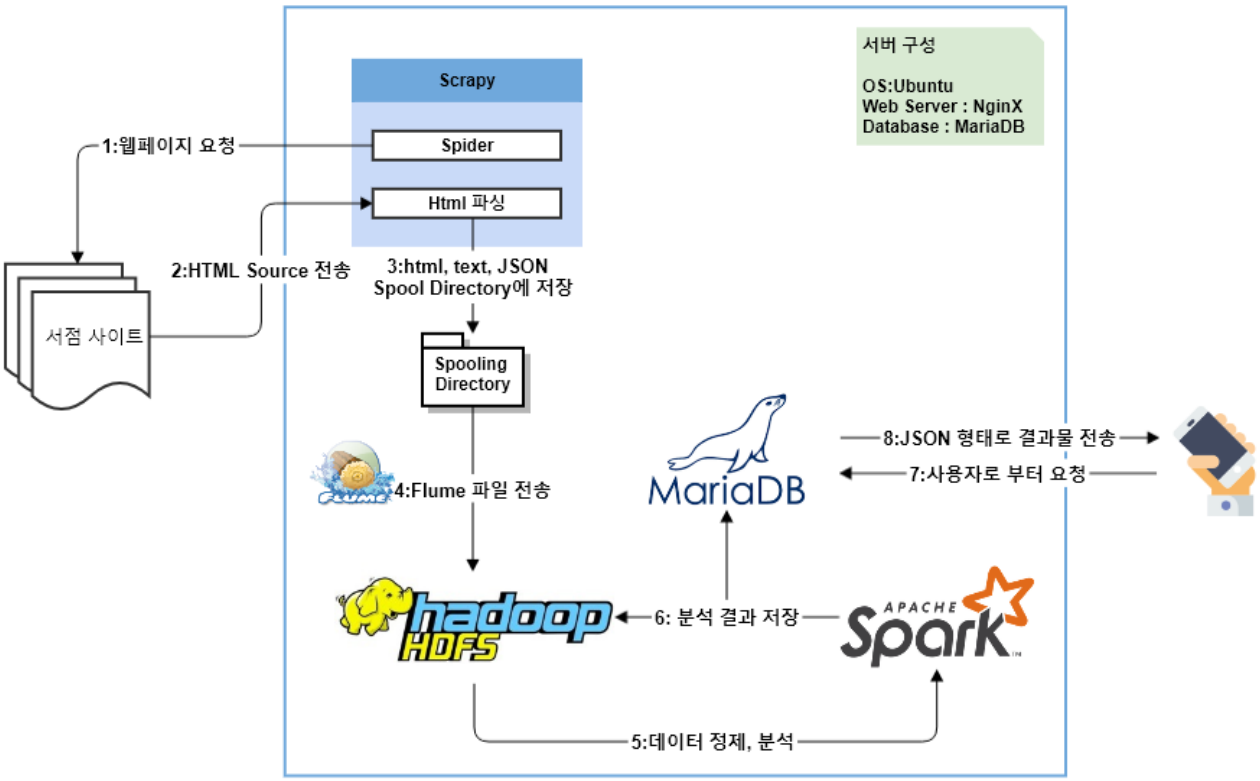
빅데이터 설계 문서

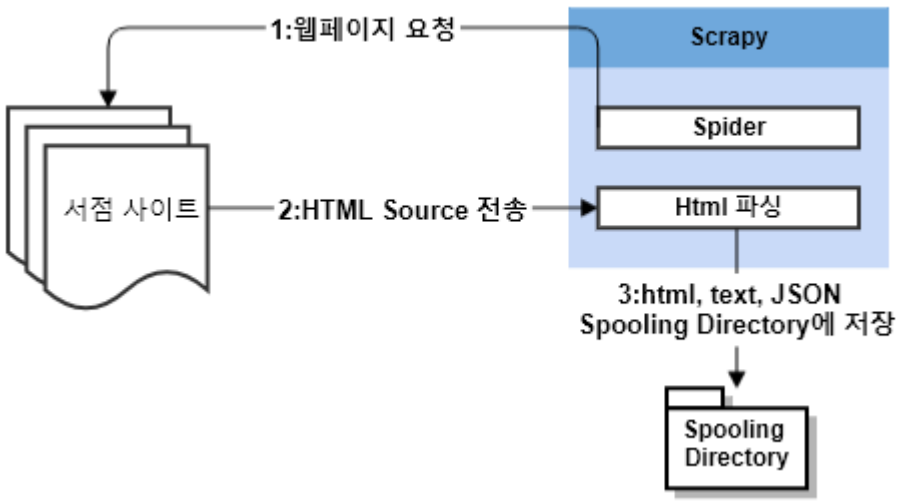
작성자 : 정새하

본 문서는 포트폴리오 MEBOOK 구현을 위해 설계한 내용을 정리한 문서 입니다

시스템 구성



데이터 수집



서점사이트 선택

- 한정된 기한 안에 많은 양의 데이터를 가져오기 위해 대형 서점 사이트를 타겟
- 도서 보유량과 리뷰 데이터가 많은 서점을 2곳 선정
- 교보문고, YES24

가져올 장르 및 데이터

- 개발 기한을 생각해 장르를 2개로 한정
- 소설, 자기계발
- 가져 올 데이터는 장르별 베스트셀러 목록, 도서 목록, 도서 정보 (제목, 저자, ISBN, 출간일, 출판사, 가격, 리뷰, 평점, 가격, 책소개, 목차 , 저자소개 등 광고를 제외한 모든 정보)

크롤러 선택

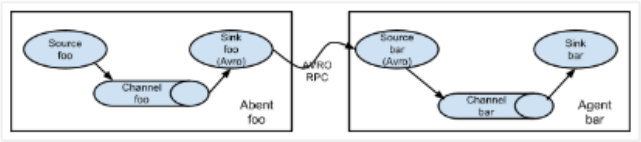
- Python 기반의 Scrapy
- Pipeline필터 기능, 로깅, Shell Mode 지원 등 데이터 수집 및 전처리에 용이
- 도메인의 깊이있는 크롤링에 특화
- 빅데이터 시장에서 가장 많이 쓰이는 언어 중 하나인 Python기반
- Python에서 제공하는 라이브러리 사용 용이
- 쉬운 설치
- 문서화 된 정보가 많아 진입하는데 어렵지 않음

데이터 저장

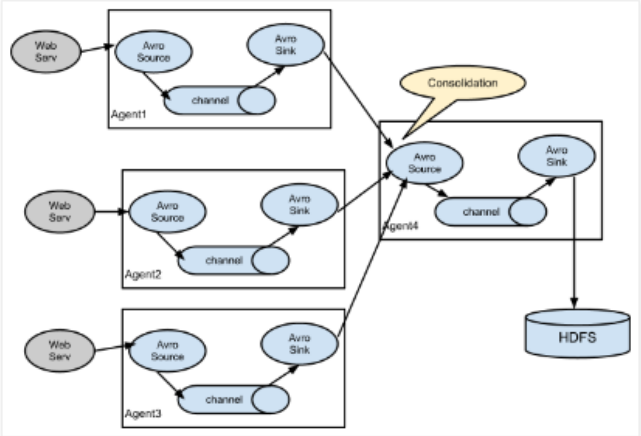
Collector(수집 모듈) 선택

- Flume
- 프로젝트 내 작업 목적과의 적합성
- Scrapy를 이용해 가져온 데이터를 HDFS에 전달, 저장 목적
- Spooling Directory를 설정하면 새로운 파일을 인식해 HDFS로 전달
- 다양한 구성이 가능하며 쉬운 확장
- Agent끼리의 연결, HDFS, HBase 등이 저장소 연결을 source, sink타입을 통해 간단하게 설정

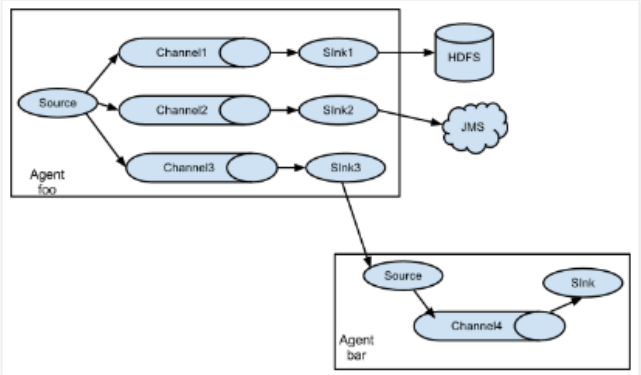
Setting multi-agent flow



Consolidation



Multiplexing the flow



- 간단하다
- Flume은 설치가 쉽고 옵션, 타입만 설정하면 데이터 저장과 전달이 가능
- 꾸준한 업데이트
- Flume은 최근 2016.10.17(설계 작성 시) 자로 업데이트가 계속 진행 중에 있다(2017.10.4 / 1.8.0 version release)
- 관심도가 꾸준히 증가하는 모듈, 많은 문서량

TRENDING

Hadoop 환경 구축

OS : Ubuntu 16.04 LTS

Hadoop Version : 2.7.3

Setting : Single Node Cluster

- Java 설치 `$sudo apt-get update`

  `$sudo apt-get install default-jdk`

  `$sudo java -version`

- SSH설치

  `$sudo apt-get install openssh-server`

  `$sudo sed -i -e 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config`

  `$sudo service ssh restart`

- 그룹 및 유저 추가

  `$sudo addgroup hadoop`

  `$sudo adduser --ingroup haddop hduser`

- Hadoop 설치

  `$wget http://apache.rediris.es/hadoop/common/hadoop-2.7.0/hadoop-2.7.0.tar.gz`

  `$sudo tar -xzvf hadoop-2.7.0.tar.gz -C /usr/local/lib/`

  `$sudo chown -R hduser:hadoop /usr/local/lib/hadoop-2.7.0`

- HDFS Directory 생성

  `$sudo mkdir -p /var/lib/hadoop/hdfs/namenode`

  `$sudo mkdir -p /var/lib/hadoop/hdfs/datanode`

  `$sudo chown -R hduser /var/lib/hadoop`

- 계정 전환

  `$sudo su -hduser`

- SSH key 생성

  `$ssh-keygen -t rsa -P ""`

  `$ssh-copy-id -i ~/.ssh/id_rsa localhost`

- Hadoop 설정
  - 자바 경로 확인

    `$readlink -f /usr/bin/java`

  - 환경 변수 수정

    `$~/.bashrc`

    `export JAVA_HOME=/usr/lib/jvm/java-8-oracle`

    `export HADOOP_INSTALL=/usr/local/lib/hadoop-2.7.0`

    `export PATH=$PATH:$HADOOP_INSTALL/bin`

    `export PATH=$PATH:$HADOOP_INSTALL/sbin`

    `export HADOOP_MAPRED_HOME=$HADOOP_INSTALL`

    `export HADOOP_COMMON_HOME=$HADOOP_INSTALL`

    `export HADOOP_HDFS_HOME=$HADOOP_INSTALL`

    `export YARN_HOME=$HADOOP_INSTALL`

    `export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native`

    `export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib/native"`

    `$source ~/.bashrc`

  - hadoop-env.sh 수정 (/usr/local/lib/hadoop-2.7.0/etc/hadoop/hadoop-env.sh)

    `export JAVA_HOME=/usr/lib/jvm/java-8-oracle`

  - core-site.xml 수정 (/usr/local/lib/hadoop-2.7.0/etc/hadoop/core-site.xml)

    `<configuration>`

    ...

    `<property>`

    `<name>fs.default.name</name>`

    `<value>hdfs://localhost:9000</value>`

    `</property>`

    ...

    `</configuration>`

  - yarn-site.xml 수정 (/usr/local/lib/hadoop-2.7.0/etc/hadoop/yarn-site.xml)

    `<configuration>`

    ...

    `<property>`

    `<name>yarn.nodemanager.aux-services</name>`

    `<value>mapreduce_shuffle</value>`

`</property>`

`<property>`

`<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>`

`<value>org.apache.hadoop.mapred.ShuffleHandler</value>`

`</property>`

...

`</configuration>`

- mapred-site.xml 생성 (/usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml)

  `$cp /usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml.template`

  `$/usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml`

- mapred-site.xml 수정 (/usr/local/lib/hadoop-2.7.0/etc/hadoop/mapred-site.xml)

  `<configuration>`

  `<property>`

  `<name>mapreduce.framework.name</name>`

  `<value>yarn</value>`

  `</property>`

  `</configuration>`

- hdfs-site.xml 수정 (/usr/local/lib/hadoop-2.7.0/etc/hadoop/hdfs-site.xml)

  `<configuration>`

  ...

  `<property>`

  `<name>dfs.replication</name>`

  `<value>1</value>`

  `</property>`

  `<property>`

  `<name>dfs.namenode.name.dir</name>`

  `<value>file:/var/lib/hadoop/hdfs/namenode</value>`

  `</property>`

  `<property>`

  `<name>dfs.datanode.data.dir</name>`

  `<value>file:/var/lib/hadoop/hdfs/datanode</value>`

  `</property>`

  ...

  `</configuration>`

- Format HDFS

  `$hdfs namenode -format`

- Hadoop 시작

  `$start-dfs.sh`

  `$start-yarn.sh`

- 구동 확인

  `$jsp`

  `Jps`

  `NodeManager`

  `NameNode`

  `ResourceManager`

  `DataNode`

  `SecondaryNameNode`

HDFS 파일 업로드 Test

HDFS 내 input 디렉토리에 README.txt 파일을 업로드 및 확인

```
hduser@saeha-virtual-machine:/usr/local/hadoop$ bin/hadoop fs -mkdir /input
17/01/13 00:39:01 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
hduser@saeha-virtual-machine:/usr/local/hadoop$ bin/hadoop fs -copyFromLocal REA
DME.txt /input
17/01/13 00:39:48 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
```

```
hduser@saeha-virtual-machine:/usr/local/hadoop$ bin/hadoop fs -ls /input
17/01/13 00:40:13 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 hduser supergroup       1366 2017-01-13 00:39 /input/README.txt
```

데이터 처리 및 분석

Spark

- MapReduce 보다 빠른 데이터 처리 및 분석
  - WordCount 처리 Test
    - HDFS내의 README.txt파일로 Word Count 실행
    - MapReduce는 Map과 Reduce 작업이 10초 이상 걸리는 반면 Spark는 단 두 줄로 1초도 안되는 속도로 WordCount 작업을 끝냄

```
scala> val testtt = sc.textFile("hdfs://localhost:9000/input/README.txt")
testtt: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/input/README.txt MapPartitionsRDD[27] at textF
ile at <console>:24

scala> val counts = testtt.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_+_);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[30] at reduceByKey at <console>:26
```

- SparkSQL 지원
- 스트리밍 처리, 기계학습(MLlib), 그래프 라이브러리(GraphX) 등의 기능 지원

참고 문헌

도서

한기용(2013). Do it! 직접 해보는 하둡 프로그래밍, 이지스퍼블리싱

정재화(2015). 시작하세요 하둡 프로그래밍, 위키북스

Hoffman, Steve.(2015). 아파치 Flume 따라잡기, 에이콘출판

백성민(2017). 스파크 2 프로그래밍, 위키북스

Frampton, Mike.(2016). *Mastering Apache Spark*, 정보문화사

Mitchell, Ryan.(2016). *Web Scraping with Python*, 한빛미디어

웹페이지

Apache Software Foundation. (2016). "Hadoop: Setting up a Single Node Cluster", https://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/SingleCluster.html (2017-01-01 방문)

VIGNESH PRAJAPATI.(2015) " How to install Apache Hadoop 2.6.0 in Ubuntu (Single node setup)" PINGAX, http://pingax.com/install-hadoop2-6-0-on-ubuntu (2017- 01 -01 방문)

CityHunter.(2016) spark install on Hadoop 알파孔의 빅데이터 놀이터,http://alphagong.blogspot.kr/2016/10/spark-201-install-on-hadoop-271_19.html (2017-01-14 방문)

Anderson Jo. (2015). "Hadoop Tutorial + IntelliJ Configuration", http://andersonjo.github.io/hadoop/2015/09/09/Hadoop-Tutorial (2017-01-20 방문)

Scrapy. (2015). "Scrapy Tutorial", https://doc.scrapy.org/en/1.0/intro/tutorial.html (2017-02-04 방문)

Apache Spark. (2016). "Quick Start", https://spark.apache.org/docs/latest/quick-start.html (2017-02-10 방문)