2.3 빅데이터 수집 실습하기

1) Crawling 실습

실습1 Python 전국 날씨 데이터 크롤링 실습하기

실습2 Python 네이버 실시간 검색어 크롤링 실습하기

실습3 Python 네이버 영화 평점·리뷰 크롤링 실습하기

2) Flume 실습

실습1 Flume 실시간(Stream) 로그 수집 실습하기

STEP_1. Apache Log 확인

ㄴ Flume 실습을 위한 로그 데이터로 실시간으로 생성되는 Apache Log를 확인(Log서버)

```
#vi /var/log/httpd/access_log
```

ㄴ Apache Log 실시간 follow 모드 확인(Log서버)

```
#tail -f /var/log/httpd/access_log
```

STEP_2. Flume 다운로드/설치/설정

ㄴ 플럼 1.9.x 버전 다운로드/압축해제/이동(Log서버, Namenode 동일)

```
#wget http://mirror.apache-kr.org/flume/1.9.x/apache-flume-1.9.x-bin.tar.gz
#tar xvfz apache-flume-1.9.x-bin.tar.gz
#mv apache-flume-1.9.x-bin /home/bigdata/
#ln -s apache-flume-1.9.x-bin flume
```

ㄴ 환경변수 설정(Log서버, Namenode 동일)

```
#vi ~/.bashrc

맨 아래 추가
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export FLUME_HOME=/home/bigdata/flume
export FLUME_CONF_DIR=/home/bigdata/flume/conf
export PATH=$PATH:$FLUME_HOME/bin
```

ㄴ 변경사항 현재 쉘에 반영(Log서버, Namenode 동일)

```
#source ~/.bashrc
```

ㄴ Log서버 Flume Agent 설정 파일 생성(Log서버)

```
#cd    /home/bigdata/flume/conf
#cp   flume-conf.properties.template      flume-agent1.conf
```

ㄴ Namenode Flume Agent 설정 파일 생성(Namenode)

```
#cd    /home/bigdata/flume/conf
#cp   flume-conf.properties.template      flume-agent2.conf
```

ㄴ Log서버 Flume Agent설정(Log서버)

```
#vi /home/bigdata/flume/conf/flume-agent1.conf

기존 내용 모두 삭제 후 아래 내용 추가
agent1.sources = execGenSrc
agent1.channels = memoryChannel
agent1.sinks = avroSink

agent1.sources.execGenSrc.type = exec
agent1.sources.execGenSrc.command = tail -f /var/log/httpd/access_log
agent1.sources.execGenSrc.batchSize = 10
agent1.sources.execGenSrc.channels = memoryChannel

agent1.sinks.avroSink.type = avro
agent1.sinks.avroSink.hostname = 192.168.xxx.101(Namenode 주소)
agent1.sinks.avroSink.port = 33333
agent1.sinks.avroSink.batch-size = 10
agent1.sinks.avroSink.channel = memoryChannel

agent1.channels.memoryChannel.type = memory
agent1.channels.memoryChannel.capacity = 100000
agent1.channels.memoryChannel.transactionCapacity = 10000
```

ㄴ Namenode Flume Agent설정(Namenode)

```
#vi /home/bigdata/flume/conf/flume-agent2.conf

기존 내용 모두 삭제 후 아래 내용 추가
agent2.sources = avroGenSrc
agent2.channels = memoryChannel
agent2.sinks = HDFS

agent2.sources.avroGenSrc.type = avro
agent2.sources.avroGenSrc.bind = 192.168.xxx.101(Namenode 주소)
agent2.sources.avroGenSrc.port = 33333
agent2.sources.avroGenSrc.channels = memoryChannel

agent2.sinks.HDFS.type = HDFS
agent2.sinks.HDFS.hdfs.path = hdfs://192.168.xxx.101:9000/flume/%Y/%m/%d
agent2.sinks.HDFS.hdfs.fileType = DataStream
agent2.sinks.HDFS.hdfs.writeFormat = text
agent2.sinks.HDFS.hdfs.batchSize = 100
agent2.sinks.HDFS.hdfs.rollSize = 0
agent2.sinks.HDFS.hdfs.rollCount = 10000
agent2.sinks.HDFS.hdfs.rollInterval = 600
agent2.sinks.HDFS.hdfs.useLocalTimeStamp = true
agent2.sinks.HDFS.channel = memoryChannel

agent2.channels.memoryChannel.type = memory
agent2.channels.memoryChannel.capacity = 100000
```

STEP_3. Flume Agent 실행

ㄴ Log서버 Flume Agent 실행(반드시 root계정으로 실행)

```
#cd /home/bigdata/flume/
#./bin/flume-ng agent --conf-file ./conf/flume-agent1.conf --name agent1
```

ㄴ Namenode Flume Agent 실행(먼저 Log서버 Agent 실행)

```
#cd /home/bigdata/flume/
#./bin/flume-ng agent --conf-file ./conf/flume-agent2.conf —name agent2

출력 로그 가운데 Creating hdfs... 메시지 확인
hdfs.BucketWriter: Creating hdfs://192.168.xxx.101:9000/flume/2020/07/10/FlumeData...

...
```

STEP_4. HDFS Browser 확인

ㄴ http://192.168.xxx.101:50070 브라우저 확인

ㄴ Utilities – Browse the file system

ㄴ /flume/2020/07/10/FlumeData.xxxxxxxxxx.tmp 파일 확인

STEP_5. HDFS 파일 내용 확인

ㄴ Agent 종료 및 HDFS 파일을 로컬로 복사 후 파일 내용 확인

```
출력 로그 가운데 Creating hdfs... 메시지 확인
hdfs.BucketWriter: Creating hdfs://192.168.xxx.101:9000/flume/2020/07/10/FlumeData...

...
[Ctrl + c] 입력해서 Namenode Flume Agent 종료

#hdfs dfs -get /flume /root
#vi /root/flume/2020/07/10/FlumeData.xxxxxxx
```

실습2 Flume 일괄(Batch) 로그 수집 실습하기

STEP_1. Flume 다운로드/설치/설정

ㄴ 플럼 1.9.x 버전 다운로드/압축해제/이동(Log서버, Namenode 동일)

```
#wget http://mirror.apache-kr.org/flume/1.9.x/apache-flume-1.9.x-bin.tar.gz
#tar xvfz apache-flume-1.9.x-bin.tar.gz
#mv apache-flume-1.9.x-bin /home/bigdata/
#ln -s apache-flume-1.9.x-bin flume
```

ㄴ 환경변수 설정(Log서버, Namenode 동일)

```
#vi ~/.bashrc

맨 아래 추가
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export FLUME_HOME=/home/bigdata/flume
export FLUME_CONF_DIR=/home/bigdata/flume/conf
export PATH=$PATH:$FLUME_HOME/bin
```

ㄴ 변경사항 현재 쉘에 반영(Log서버, Namenode 동일)

```
#source ~/.bashrc
```

ㄴ Log서버 Flume Agent 설정 파일 생성(Log서버)

```
#cd    /home/bigdata/flume/conf
#cp   flume-conf.properties.template      flume-agent1-spooldir.conf
```

ㄴ Namenode Flume Agent 설정 파일 생성(Namenode)

```
#cd    /home/bigdata/flume/conf
#cp   flume-conf.properties.template      flume-agent2-spooldir.conf
```

ㄴ Flume Agent가 로그파일을 수집할 대상 디렉터리 생성(Log서버)

```
#mkdir /home/bigdata/working/logs
```

ㄴ Log서버 Flume Agent설정(Log서버)

```
#vi /home/bigdata/flume/conf/flume-agent1-spooldir.conf

기존 내용 모두 삭제 후 아래 내용 추가
agent1.sources = spoolDirSource
agent1.channels = spoolDirChannel
agent1.sinks = avroSink

agent1.sources.spoolDirSource.type = spooldir
agent1.sources.spoolDirSource.channels = spoolDirChannel
agent1.sources.spoolDirSource.spoolDir = /home/bigdata/working/logs
agent1.sources.spoolDirSource.deletePolicy = immediate

agent1.sinks.avroSink.type = avro
agent1.sinks.avroSink.channel = spoolDirChannel
agent1.sinks.avroSink.hostname = 192.168.xxx.101(Namenode 주소)
agent1.sinks.avroSink.port = 4545

agent1.channels.spoolDirChannel.type = memory
agent1.channels.spoolDirChannel.capacity = 100
```

ㄴ Namenode Flume Agent설정(Namenode)

```
#vi /home/bigdata/flume/conf/flume-agent2-spooldir.conf

기존 내용 모두 삭제 후 아래 내용 추가
agent2.sources = targetSource
agent2.channels = targetChannel
agent2.sinks = targetSink

agent2.sources.targetSource.type = avro
agent2.sources.targetSource.channels = targetChannel
agent2.sources.targetSource.bind = 192.168.100.101
agent2.sources.targetSource.port = 4545

agent2.sinks.targetSink.type = hdfs
agent2.sinks.targetSink.channel = targetChannel
agent2.sinks.targetSink.callTimeout = 15000
agent2.sinks.targetSink.hdfs.path = hdfs://192.168.100.101:9000/flume/logs/%Y-%m-%d/
agent2.sinks.targetSink.hdfs.fileType = DataStream
agent2.sinks.targetSink.hdfs.writeFormat = Text
agent2.sinks.targetSink.hdfs.useLocalTimeStamp = true
agent2.sinks.targetSink.hdfs.filePrefix = access_log
agent2.sinks.targetSink.hdfs.fileSuffix = .log
agent2.sinks.targetSink.hdfs.threadsPoolSize = 10

agent2.channels.targetChannel.type = memory
agent2.channels.targetChannel.capacity = 100
```

STEP_2. Flume Agent 실행

ㄴ Log서버 Flume Agent 실행(반드시 root계정으로 실행)

```
#cd /home/bigdata/flume/
#./bin/flume-ng agent --conf-file ./conf/flume-agent1-spooldir.conf --name agent1
…
```

ㄴ Namenode Flume Agent 실행(먼저 Log서버 Agent 실행)

```
#cd /home/bigdata/flume/
#./bin/flume-ng agent --conf-file ./conf/flume-agent2-spooldir.conf --name agent2
…
```

STEP_3. 로그 파일 수집 확인

ㄴ FileZilla FTP로 Log서버 접속 후 /home/bigdata/working/logs에 로그 파일 업로드

```
출력 로그 가운데 Creating hdfs... 메시지 확인
hdfs.BucketWriter: Creating hdfs://192.168.xxx.101:9000/flume/2020-07-10/access_log…

…
```

STEP_4. HDFS Browser 확인

ㄴ http://192.168.xxx.101:50070 브라우저 확인

ㄴ Utilities – Browse the file system

ㄴ /flume/logs/2020-07-10/access_log.xxxxxxxx.log 파일 확인

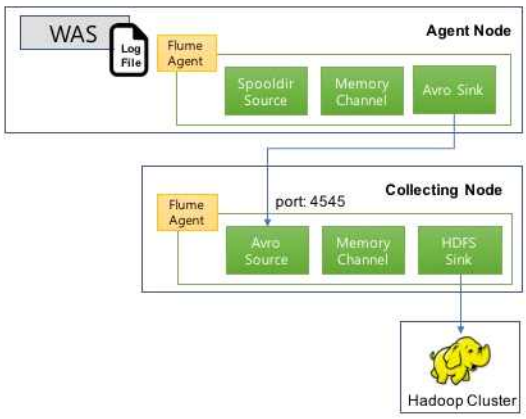
STEP_5. HDFS 파일 내용 확인

ㄴ Agent 종료 및 HDFS 파일을 로컬로 복사 후 파일 내용 확인

```
hdfs.BucketWriter: Creating hdfs://192.168.xxx.101:9000/flume/2020-07-10/access_log…

…
[Ctrl + c] 입력해서 Namenode Flume Agent 종료

#hdfs dfs -get /flume/logs /root
#vi /root/flume/logs/2020-07-10/access_log.xxxxxxxxx.log
```



3) Sqoop 실습

실습1 Sqoop 설치 실습하기

STEP_1. Sqoop 설치/설정

ㄴ Hadoop 2.x 버전일 경우 sqoop-1.4.x.bin__hadoop-2.x.tar.gz 다운로드 해야됨

ㄴ Sqoop 다운로드/압축해제/이동/링크 생성(Namenode 실행)

```
#wget http://mirror.apache-kr.org/sqoop/1.4.x/sqoop-1.4.x.bin__hadoop-2.x.tar.gz
#tar zxvf sqoop-1.4.x.bin__hadoop-2.x.tar.gz
#mv sqoop-1.4.x.bin__hadoop-2.x /home/bigdata/
#cd /home/bigdata/
#ln -s sqoop-1.4.x.bin__hadoop-2.x sqoop
```

ㄴ Sqoop 환경변수 설정(Namenode 실행)

```
#vi ~/.bashrc

맨 아래 추가
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export SQOOP_HOME=/home/bigdata/sqoop
export SQOOP_CONF_DIR=/home/bigdata/sqoop/conf
export PATH=$PATH:$SQOOP_HOME/bin
```

ㄴ 변경사항 현재 쉘에 반영(Namenode 실행)

```
#source ~/.bashrc
```

STEP_2. RDBMS 드라이버 설치(Namenode 실행)

ㄴ http://dev.mysql.com/downloads/connector/j/ 에서 리눅스용 mysql-connector-java-5.x.tar.gz 파일 다운로드

ㄴ FileZilla FTP로 mysql-connector-java-5.x.tar.gz 파일을 /root 경로에 업로드

ㄴ mysql-connector-java-5.x.tar.gz 압축해제/라이브러리 복사

```
#tar zxvf mysql-connector-java-5.x.tar.gz
#cp mysql-co...-java-5.x/mysql-connector-java-5.x-bin.jar $SQOOP_HOME/lib
```

STEP_3. RDBMS Sqoop 계정 생성 및 테스트용 database 생성(Namenode 실행)

ㄴ mysql(mariadb)가 설치되어 있어야 함(메뉴얼 참고)

```
#mysql -u root -p

mysql>CREATE USER 'sqoop'@'localhost' identified by '1234';
mysql>CREATE USER 'sqoop'@'%' identified by '1234';
mysql>CREATE DATABASE sqoop;
mysql>GRANT ALL PRIVILEGES ON sqoop.* TO 'sqoop'@'localhost';
mysql>FLUSH PRIVILEGES;
mysql>exit
```

STEP_4. RDBMS 테스트용 TABLE 생성 데이터 입력(Namenode 실행)
ㄴ mysql(mariadb)가 설치 되어 있어야 함(메뉴얼 참고)

```
#mysql -u sqoop -p
mysql>use sqoop;

Import용 테이블 생성
mysql>CREATE TABLE User1 (uid VARCHAR(10),
                          name VARCHAR(10), hp CHAR(13), age INT);

Sample 데이터 입력
mysql>INSERT INTO User1 VALUES ('A101', '김유신', '010-1234-1111', 23);
mysql>INSERT INTO User1 VALUES ('A102', '김춘추', '010-1234-2222', 21);
mysql>INSERT INTO User1 VALUES ('A103', '이순신', '010-1234-3333', 35);

Export용 테이블 생성
mysql>CREATE TABLE User2 LIKE User1;
mysql>exit
```

실습2 Sqoop Import 실습하기(RDBMS → HDFS로 저장)
STEP_1. Sqoop import 실행
ㄴ Hadoop 실행, 한 줄로 입력, 한 칸씩 띄어쓰기, 복사/붙여넣기 하지 말고 직접 입력

```
#sqoop import --connect jdbc:mysql://192.168.xxx.xxx:3306/sqoop --table User1
 --username sqoop --password 1234 -m 1
 --target-dir hdfs://192.168.xxx.xxx:9000/sqoop/User1
…
출력 로그 중 Job ~ completed successfully 메시지 확인
mapreduce.Job: Job job_1594343546908_0001 completed successfully
```

STEP_2. HDFS 확인
ㄴ http://192.168.xxx.101:50070 브라우저 확인
ㄴ Utilities – Browse the file system - /sqoop/User1 확인 및 파일 내용 확인

실습3 Sqoop Export 실습하기(HDFS → RDBMS로 저장)
STEP_1. Sqoop export 실행
ㄴ Hadoop 실행, 한 줄로 입력, 한 칸씩 띄어쓰기, 복사/붙여넣기 하지 말고 직접 입력

```
#sqoop export --connect jdbc:mysql://192.168.xxx.101:3306/sqoop --table User2
 --export-dir /sqoop/User1 --username sqoop --password 1234 -m 1
…
출력 로그 중 Job ~ completed successfully 메시지 확인
mapreduce.Job: Job job_1594343546908_0001 completed successfully
```

STEP_2. RDBMS 확인
ㄴ HeidiSQL 접속 – sqoop 데이터베이스 – User2 테이블 데이터 확인