비정형 빅데이터 수집 모듈의 구현 및 비교

*김정기, 천요섭, 김우생
광운대학교 컴퓨터소프트웨어학과
e-mail : kjk135246@naver.com

Implementation and Comparison of Atypical Big-Data Collecting Modules

*JungKi Kim, YoSeop Cheon, WooSaeng Kim
School of Compuer Software
KwangWoon University

요 약

최근 스마트폰의 보급으로 블로그, SNS 등에서 방대한 양의 데이터가 발생함에 따라 이를 수집하고 분석하는 작업의 중요성이 커지고 있다. 이러한 데이터는 크게 정형 데이터와 비정형 데이터로 나눌 수 있는데, 특히 비정형 데이터는 전체 데이터의 약 80%를 차지할 정도로 그 양과 가치가 매우 크다. 이 논문에서는 빅데이터 환경에서 발생하는 이러한 비정형 데이터를 수집하는 모듈 중 가장 널리 알려진 Chukwa와 Flume에 대한 개발 및 비교 분석을 시도 하였다.

1. 서론

최근 스마트폰 등 모바일 기기의 대중화로 블로그, SNS 등에서 방대한 양의 데이터가 발생하고 있다. 이렇게 발생하는 데이터에는 다양한 정보들이 포함되어 있으며, 이러한 다양한 형태의 대용량 데이터를 처리하고 관리하는 것의 중요성이 부각되면서 빅데이터 수집과 분석에 대한 요구가 급증하였다. 이에 빅데이터에 대한 많은 연구가 진행됨과 동시에 빅데이터를 영화 추천 서비스, 영화 흥행 예측, 주식 예측 등 다양한 분야에 활용하고 있다.

비정형 데이터는 빅데이터 시대의 등장에 큰 영향을 미친 데이터로써, 텍스트 데이터뿐만 아니라 각종 이미지, 음성, 동영상 등 형식 없이 저장되는 데이터를 말한다. 형식이 없기 때문에 분석에 많은 어려움을 가지고 있지만 전체 데이터의 약 80%를 차지하기 때문에 그 가치를 무시할 수 없다. 특히 블로그, SNS 등을 통해 비정형 데이터가 늘어나는 속도가 매우 빠르기 때문에 비정형 데이터 비중은 더욱 커질 것이다.

본 논문에서는 빅데이터 환경에서 발생하는 비정형 데이터를 수집하는 모듈 중 가장 널리 알려진 Chukwa와 Flume에 대한 개발 및 성능 분석을 하여 각각의 모듈에 대한 장단점을 기술하고자 한다[1,2].

본 논문의 구성은 다음과 같다. 2장에서는 전체적인 하둡 에코시스템(Hadoop Ecosystem)의 구성 및 각각의 모듈을 소개한다. 3장에서는 Chukwa와 Flume에 대한 시스템 설계를 설명한다. 4장에서는 두 모듈의 구현과 성능 분석 등을 통한 기능 비교를 한다. 마지막 5장에서는 본 논문에 대한 결론과 향후 연구 방향에 대하여 논의 한다.

2. 관련 연구

하둡 에코시스템은 아래의 그림 1과 같이 구성된다[3,4]. 정형 데이터 수집 모듈인 Sqoop은 대용량 데이터 전송 솔루션이며, HDFS, RDBMS, DW, NoSQL 등 다양한 저장소에 대용량 데이터를 신속하게 전송할 수 있는 방법을 제공한다. Hiho는 Sqoop과 같은 대용량 데이터 전송 솔루션이며, Hadoop에서 데이터를 가져오기 위한 SQL을 지정할 수 있으며, JDBC 인터페이스를 지원한다. 결과 데이터 액세스 모듈인 HBase는 HDFS 기반의 비관계형 분산 데이터 베이스이다. 실시간 랜덤 조회

및 업데이트가 가능하고 각각의 프로세스들은 개인의 데이터를 비동기적으로 업데이트할 수 있다. 데이터 저장 모듈인 HDFS는 마스터, 슬레이브 구조의 분산파일 시스템이고 데이터 처리 모듈인 MapReduce는 분산환경에서 병렬처리를 하기 위한 시스템이다. Pig는 대용량 데이터 집합을 분석하기 위한 플랫폼으로 높은 수준의 스크립트 언어로 구성되어 있다. Hive는 Hadoop에서 동작하는 데이터 웨어하우스 인프라 구조로서 데이터 요약, 질의, 분석 기능을 제공한다. 분산 코디네이터인 Zoopeeker는 분산 환경에서 서버들간에 상호 조정이 필요한 다양한 서비스를 제공하는 시스템이다. Avro는 데이터 직렬화를 지원하는 프레임워크이다. 데이터 분석과 마이닝에 사용되는 Mahout은 Hadoop 기반으로 데이터 마이닝 알고리즘을 구현한 오픈 소스이다. 워크 플로우 관리 모듈인 Oozie는 Hadoop 작업을 관리하는 워크 플로우 스케줄러 시스템으로 Map Reduce 작업이나 Pig 작업 같은 특화된 액션들로 구성된 워크 플로우를 제어한다.

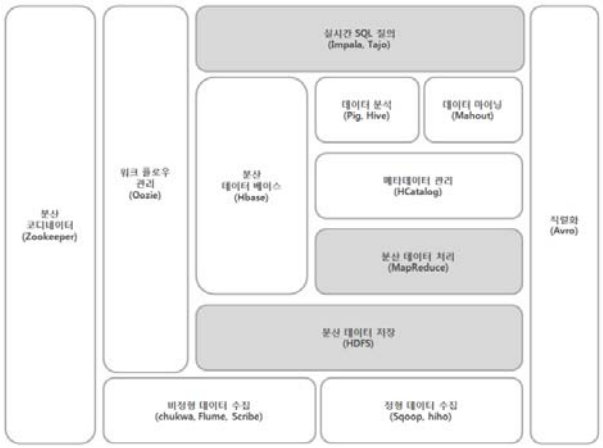

그림 1. Hadoop Ecosystem

3. 설계

단일 머신에서의 Flume의 구조적 설계는 그림2와 같다. 하나의 Agent로 구성 된 데이터 플로우는 이벤트[1]를 전송하고 수집하는 일련의 논리 노드(Source, Channel, Sink)들로 구성한다. Agent의 Source가 로그생성기로 생성한 파일로부터 이벤트를 받으면 Channel에 이벤트를 저장하고, Sink는 Channel로부터 이벤트를 HDFS로 보내고 Channel에서 제거한다. 이 때 Source와 Sink는 비동기적으로 실행된다.

1) 이벤트는 Flume의 Agent를 통해 흐르는 데이터의 단위를 나타낸다.

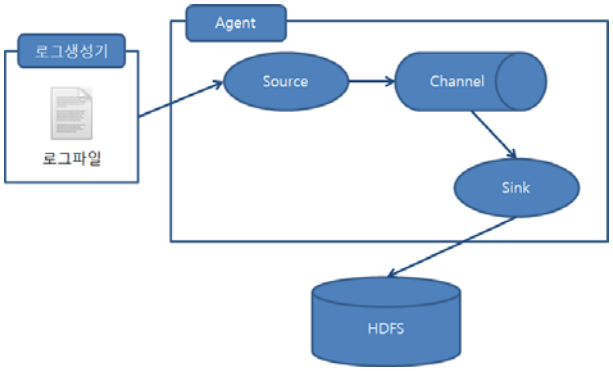

그림 2. Flume 설계

단일 머신에서 Chukwa의 구조적 설계는 그림 3과 같다. Chukwa의 Adaptor, Agent, Collector는 각각 하나씩으로 구성한다. Adaptor는 로그생성기로 생성한 파일로부터 로그를 수집한다. Adaptor에서 수집된 로그는 Agent에 의해 메타데이터가 포함된 일련의 바이트인 chunk 단위로 Collector로 전송되고 이는 다시 Collector에 의해 HDFS에 기록된다.

그림 3.Chukwa 설계

4. Flume과 Chukwa 구현 및 성능 비교

4.1 Flume의 구현

본 논문에서는 Apache Flume 1.4.0 버전을 사용하였다. Flume의 실행을 위해서 먼저 Flume이 설치된 디렉토리에 있는 conf 디렉토리 내의 환경설정파일을 작성한다. 여기서 환경설정 파일은 <파일이름>.conf의 형식으로 만든다. 본 논문에서는 ex.conf으로 하였으며 아래의 표1과 같이 작성하였다. 본 논문에서는 로그파일을 수집하기 위해 Source의 타입을 exec로 하였다. exec는 Flume의 Agent가 리눅스 쉘명령어를 주기적으로 실행하여 그 결과를 수집하게 되는 속성값이며, 여기에 해당하는 쉘명령어는 command의 속성값에서 설정한다. 여기서의 command의 속성값은 demo 로그파일을 갱신하면서 출력하는 리눅스 명령어인 tail -F를 사용하였다. Sink의 타입은 hdfs로 하여 수집한 데이터를 HDFS로 저장하도록 하였다. Flume의 Agent 데몬을 실행할 때는 터미널을 실행한 다음 Flume이 설치된 디렉토리로 이동하여 커맨드 라인에서 <환경설정 파일의 디렉토리>, <환경설정 파일이름>, <실행할 에이전트의 이름>을 입력한다. 본 논문에서는 Flume Agent

데몬을 실행하기 위한 명령어로 다음을 사용하였다. “bin/flume-ng agent --conf conf --conf-file conf/ex.conf --name a1 -Dflume.root.logger=INFO,console."

표 1. Flume 환경설정 (example.conf)

```
# Name the components on this agent
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = exec
a1.sources.r1.command = tail -F /tmp/ws_demo_log/demo

# Describe the sink
a1.sinks.k1.type = hdfs
a1.sinks.k1.hdfs.path
 = hdfs://localhost:9000/flume/events2

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

4.2 Chukwa의 구현

본 논문에서는 Apache Chukwa 0.4.0 버전을 사용하였고 단일머신으로 구성했다. Chukwa의 실행을 위해서 먼저 Chukwa가 설치된 디렉토리에 있는 conf 디렉토리 내의 환경설정파일을 작성한다. 표2의 내용은 Chukwa에서 데이터 수집을 위한 Adaptor를 설정하는 환경설정 파일로 conf/Initial_adaptors에 데이터를 수집할 Adaptor를 설정한다. Adaptor는 필요에 따라 변경하여 사용할 수 있으며 본 연구에서는 한 파일에서 로그를 지속적으로 생성했으므로 한 파일을 반복적으로 수집하는 CharFileTailingAdaptorUTF8NewLineEscaped를 사용한다[2]. 그리고 conf/agents 와 conf/collectors 에는 각각 agent 와 collector의 목록을 입력하는데 단일머신이므로 localhost를 입력한다. Chukwa의 실행을 위해서는 로그를 수집하는 Agent와 수집된 로그를 모아서 기록하는 Collector를 따로 실행 시켜야 한다. bin 디렉토리의 start-agents.sh로 Agent를 실행하고 start-collectors.sh로 Collector를 실행할 수 있다. chukwa가 설치된 디렉토리에서 bin/start-agents.sh와 같이 실행한다. 수집된 로그는 HDFS에 저장된다.

표 2. Chukwa 환경설정

```
add
org.apache.hadoop.chukwa.datacollection.adaptor.filetailer.C
harFileTailingAdaptorUTF8NewLineEscaped
S 0 /tmp/ws_demo_log/demo 0
```

4.3 성능 및 기능 비교

본 논문에서는 Flume과 Chukwa의 성능을 비교 분석하기 위하여 단일머신에서 직접 로그를 생성하여 각각의 모듈이 로그를 수집하여 HDFS에 저장하도록 구성했다. 테스트는 Intel Core 2 Duo E4500의 CPU, 3G인 메모리, OS는 Ubuntu 12.04 LTS 64bit인 환경에서 수행하였으며, Hadoop, Flume, Chukwa의 버전은 각각 1.0.3, 1.4.0, 0.4.0이다. 로그는 558Kb/sec의 속도로 지속적으로 생성하였으며 각 모듈은 5분씩 실행하였다. 측정은 리눅스 명령어 top를 사용하여 30초 간격으로 CPU점유율과 RAM점유율을 측정하였고, HDFS내에 각각 생성된 데이터의 크기를 측정하였다. 표 3을 보면 Flume이 Chukwa에 비해 많은 수치의 CPU 점유율과 메모리 점유율을 보였으나 생성한 로그파일의 크기는 현저히 작음을 보였다. 로그 파일을 읽어 HDFS로 데이터를 생성하는데 있어서 Flume은 Chukwa보다 매우 낮은 효율을 보임을 알 수 있다.

표 3. 성능 비교

	Flume	Chukwa (Agent)	Chukwa (Collector)
CPU max	55%	3%	2%
CPU min	33%	0%	0%
CPU avg	45%	1%	1%
RAM max	3.8%	1.6%	2.6%
RAM min	3.6%	1.4%	2.4%
RAM avg	3.7%	1.5%	2.5%
Output size	3.88MB	19.08MB	

표4는 두 모듈의 기능을 비교한 내용이며 Flume은 소셜, 이메일, 트래픽 등 다양한 소스로부터 대량의 로그 데이터를 효과적으로 수집한다. 장애에 쉽게 대처 가능하며 신뢰성 있는 서비스를 제공하며 간단하게 확장 가능하다. 이러한 이유로 빅데이터 수집 부분에서 가장 많이 사용된다. Chukwa는 HDFS의 장점을 모두 상속하며, HICC로 웹-포탈 형식으로 데이터를 확인 가능하다. 하지만 Hadoop에 너무 의존적이라는 단점이 있다. 앞서 성능을 비교했을 때는 Chukwa가 Flume보

다 훨씬 높은 효율을 보였다. 하지만 Flume은 다양한 소스로부터 로그를 수집할 수 있다는 장점을 가진다.

표 4. 성능 비교

	Apache Flume	Apache Chukwa
Overview	분산 환경에서 로그 데이터를 신뢰성, 확장성을 바탕으로 효율적으로 수집, 전송 가능한 서비스.	분산 데이터 수집 및 출력, 모니터링, 분석하는 서비스를 제공하는 시스템
Home	http://flume.apache.org/	http://chukwa.apache.org/
Last Version	1.4.0 (2013.07.02)	0.5.0 (2012.01.26)
Status	Apache Top-Level Project	Apache Incubator Project

5. 결론 및 향후 연구

본 논문에서는 비정형 데이터를 수집하는 모듈 중 Chukwa와 Flume에 대하여 설계 및 개발하고 두 모듈의 장단점 비교 및 분석을 하였다. Chukwa와 Flume 이외에도 다양한 오픈 소스로 된 데이터 수집 모듈들이 있다. 하지만 Java 언어로 개발된 Chukwa와 Flume은 간결한 아키텍처를 제공함으로써 유지보수성과 무한한 확장 가능성을 가지고 여러 가지 제약을 자유롭게 풀어나갈 수 있는 수단을 제공한다는 점에서 장점을 가진다. 향후에는 두 모듈을 이용해 수집한 비정형 데이터를 분석해보고자 한다.

감사의 글

본 논문은 2014년도 KWIX(KwangWoon IT Exhibition)로 지원 되었습니다.

참고문헌

[1] http://flume.apache.org/FlumeUserGuide.html

[2] http://wiki.apache.org/hadoop/Chukwa

[3] 정재화, “시작하세요. 하둡 프로그래밍”, 위키북스, 2012.

[4] 한기용, “Do it 직접해보는 하둡 프로그래밍”, 이지스퍼블리싱, 2012.