


# Hadoop echo system

하둡 소프트웨어 라이브러리는 간단한 프로그래밍 모델을 사용하여 여러대의 컴퓨터 클러스터에서 대규모 데이터 세트를 분산 처리 할 수 있게 해주는 프레임워크  
단일 서버에서 수천대의 머신으로 확장 할 수 있도록 설계됨  
일반적으로 하둡파일시스템(HDFS)과 맵리듀스(MapReduce)프레임워크로 시작되었으나,  
여러 데이터저장, 실행엔진, 프로그래밍 및 데이터처리 같은 하둡 생태계 전반을 포함하는  
의미로 확장 발전 되었다.


하둡 코어 프로젝트: HDFS(분산데이터 저장), MapReduce(분산처리)

하둡 서브 프로젝트: 나머지 프로젝트들 -> 데이터 마이닝, 수집, 분석 등을 수행

분야	솔루션
NoSQL	Hbase, Cassandra, MongoDB, CouchDB, Couchbase, Cloudata, Riak, Neo4j
Cache	Redis, Memcachde
RPC	Thrift, Avro, Protocol Buffer
Collect	Scribe, Flume, Chukwa, Logstash, Fluentd
Query	Hive, Pig, Hcatalog, Impala, Tajo, SparkSQL,, BigQuery
Streaming	Akka, Storm, SparkStreaming, Esper, S4
Search	Elastic Search, Solr, Katta
File System	Hadoop, Swift, GlusterFS, Ceph
ETC	Machine Learning(Mahout), Distributed Coordinato(Zookkkper), Queue(Kafka), Data Intergration(Sqoop), Statistis(R), Workflow(Oozie)




**HUE**  
Hue is a Web interface for analyzing data with Apache Hadoop




**Apache Ambari**  
<http://incubator.apache.org/ambari>


**Ambari**  
Provisioning, Managing and Monitoring




**Apache Kafka**  
Kafka  
A high-throughput distributed messaging system




**Oozie**  
Workflow Engine




**Apache Zeppelin**  
A web-based notebook that enables interactive data analytic




**Flink**  
distributed stream and batch data processing




**Spark**  
Lightning-Fast Cluster Computing




**Scoop**  
Data Exchange




**Mahout**  
Data Mining




**DRILL**  
Real-time SQL query




**TAJO**  
Real time SQL query




**AVRO**  
data serialization system




**PIG**  
Scripting




**HIVE**  
SQL-Query




**IMPALA**  
Real-time SQL query




**TEZ**  
Runtime Engine




**HBase**  
Columnar Store




**Cassandra**  
Distributed storage system




**Redis**  
in-memory data structure store




**Chukwa**  
Log Collector




**Zookeeper**  
Coordination




**MESOS**  
open-source cluster manager




**YARN**  
Hadoop Distributed File System



**MapReduce**  
Distributed Processing Framework



**Flume**  
Log Collector



**HDFS**  
Hadoop Distributed File System

# 1. 데이터 수집

## Unstructured Data

Apache flume, Facebook Scribe, Apache Chukwa, Netflix suro

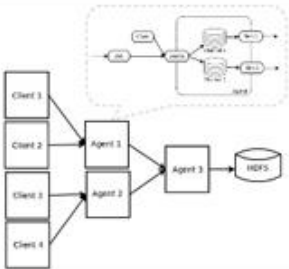
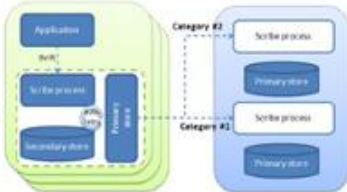
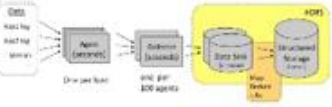
## Structured Data

Hiho

Apache Sqoop

### 대용량 고속 이벤트 데이터(로그) 수집

- 확장성 : 수집대상 서버는 무한대로 확장된다. 수집에서 수천, 수만 대로 수집대상 서버는 늘어 날 것이다.
- 안정성 : 수집되는 데이터가 유실되지 않고 안정적으로 저장되어야 한다.
- 유연성 : 다양한 포맷의 데이터, 다양한 프로토콜을 지원해야 한다.
- 실시간성 : 수집된 데이터를 실시간으로 반영해야 한다.

구분	Apache Flume	Facebook Scribe	Apache Chukwa
개요	대용량의 로그 데이터를 분산,안정성,가용성을 바탕으로 효율적으로 수집,집계,이동이 가능한 로그수집 솔루션	Facebook이 개발하여 오픈소스화한 로그수집서버. 대량의 서버로 부터 실시간으로 스트리밍 로그 수집을 위한 솔루션	Apache Hadoop의 서브 프로젝트로 분산되어 있는 서버에서 로그 데이터를 수집, 저장, 분석하기 위한 솔루션
Home	<a href="http://flume.apache.org/">http://flume.apache.org/</a>	<a href="https://github.com/facebook/scribe">https://github.com/facebook/scribe</a>	<a href="http://chukwa.apache.org/">http://chukwa.apache.org/</a>
최종버전	1.4.0 (2013.7.2)	2.2(2010)	0.5.0 (2012.1.26)
현재	Cloudera -> Apache Top-Level Project	Facebook -> Open Source	Yahoo -> Apache Incubator project
WIKI	<a href="https://cwiki.apache.org/confluence/display/FLUME/Home">https://cwiki.apache.org/confluence/display/FLUME/Home</a>	<a href="https://github.com/facebook/scribe/wiki">https://github.com/facebook/scribe/wiki</a>	<a href="http://wiki.apache.org/hadoop/Chukwa/">http://wiki.apache.org/hadoop/Chukwa/</a>
문서화	풍부	빈약	풍부
구현언어	Java	C++	Java
커뮤니티	활발	활발	활발
요약	다양한 소스로 부터 데이터를 수집하여 다양한 방식으로 데이터를 전송이 가능하지만, 아키텍처가 단순하고 유연하며 확장 가능한 데이터 모델을 제공하여, 실시간 분석 Application을 쉽게 개발 수집부분의 기반이 되는 솔루션. 각종 Source, Sink 등 제공으로 쉽게 확장 가능	Facebook의 자체 Scaling 작업을 위해 설계되어 현재 매일 수백 억건의 메시지를 처리하고 있다. 클라이언트 서버의 타입에 상관없이 다양한 방식으로 로그를 읽어 들일수 있다. 단, Apache Thrift는 필수. Thrift 기반 Scribe API를 활용하여 확장 가능	수집된 로그 파일을 HDFS에 저장한다. HDFS의 장점을 그대로 수용하고 실시간 분석도 가능하다. 반면에 Hadoop에 너무 의존적이라는 단점도 있다.
Outline			

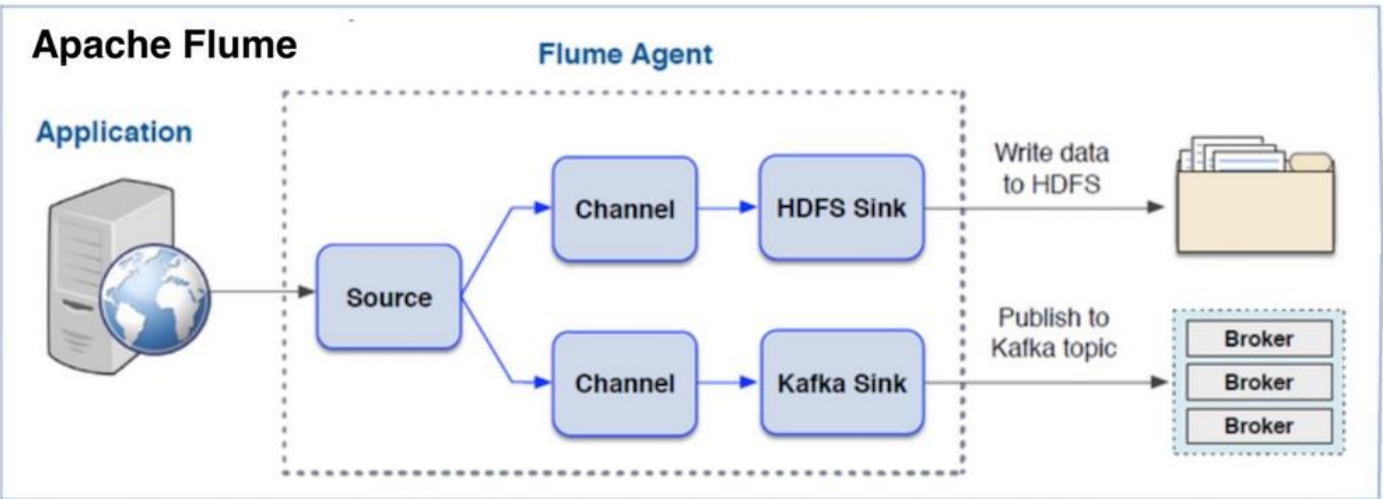
## 1-1. 데이터 수집 - Flume

**Flume(플럼)** - 오픈소스 프로젝트 개발 로그수집 기술으로 여러 서버에서 생산된 대용량 로그 데이터를 효과적으로 수집하여, HDFS과 같은 원격 목적지에 데이터를 전송하는 기능  
구조가 단순, 유연하여 다양한 유형의 스트리밍 데이터 플로우(Streaming Data Flow)아키텍처를 구성할 수 있음

**Reliability(신뢰성)** : 장애가 나더라도 로그, 이벤트를 유실 없이 전송함을 보장하도록 설계

**Scalability(확장성)** : 수평확장(Scale-Out)이 가능하여 분산수집이 가능한 구조로 설계

Flume은 Event, Agent 의 개념 Agent는 Source, Channel, Sink로 구성되어 있음

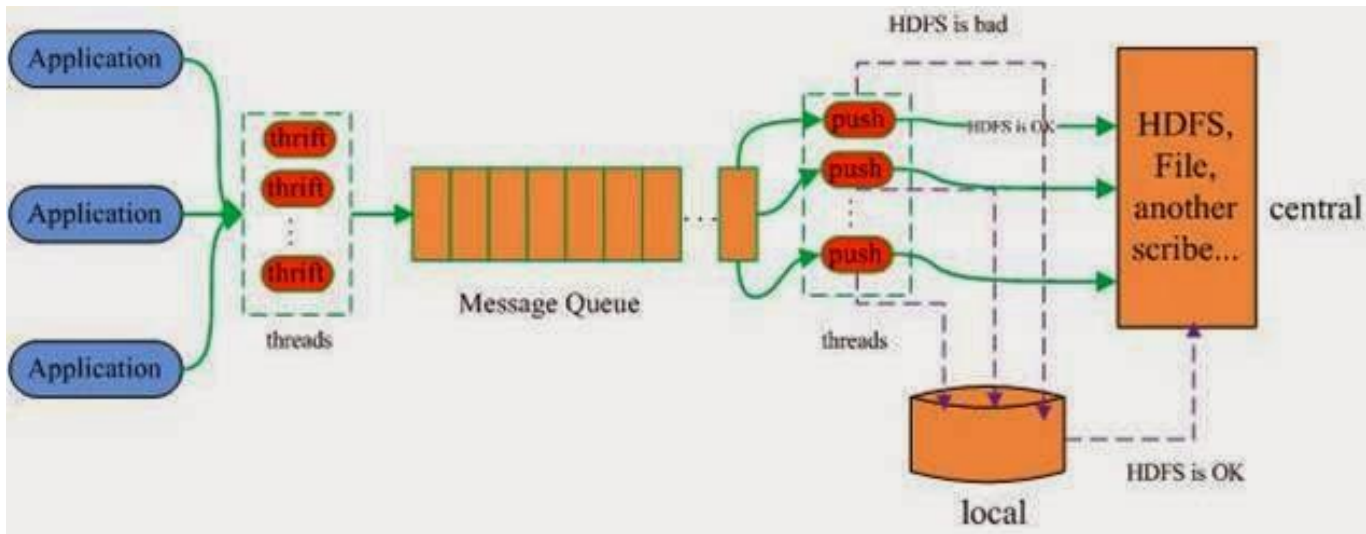


## 1-2. 데이터 수집 - Facebook scribe

**Scribe(스크라이브)** - 스크라이브(Scribe)는 수많은 서버로부터 실시간으로 스트리밍 되는 로그 데이터를 집약시키기 위한 서버. 클라이언트 사이트의 수정 없이 스케일링 가능하고 확장 가능하도록, 또 임의의 머신이나 네트워크의 실패에 대해 안전하도록 설계 됨

스크라이브 서버들은 방향 그래프로 정렬되며 각 서버는 그래프의 다음 서버에 관해서만 인지하고 있다. 이 네트워크 토폴로지는 서버가 커짐에 따라 팻인(fat-in) 추가 계층을 추가하는 것, 그리고 데이터센터 간에 메시지를 보내기 전에 메시지를 일괄 처리하는 것을 허용하며 이를 위해 데이터센터 토폴로지의 명시적인 이해가 필요한 코드 없이 단순한 구성만 수반된다.

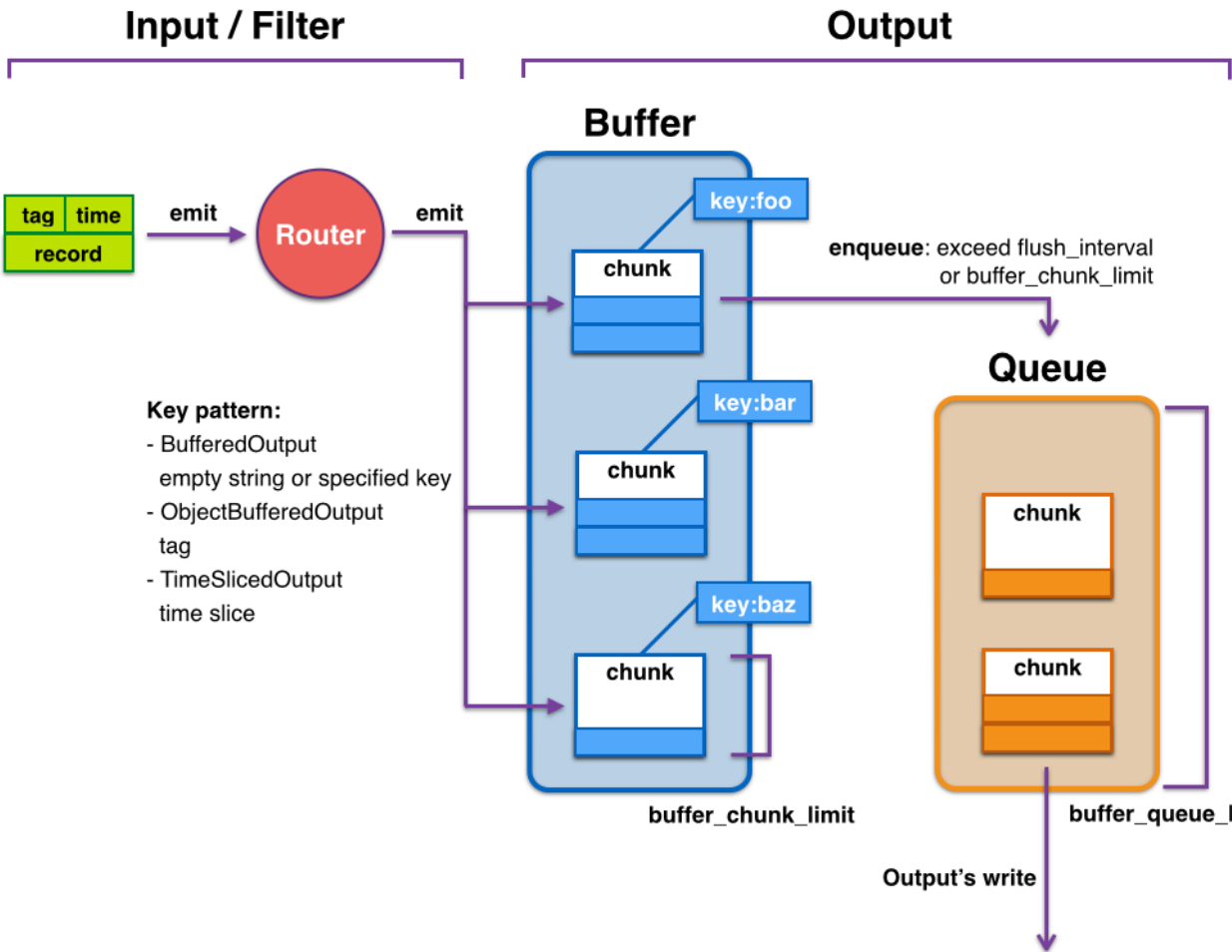
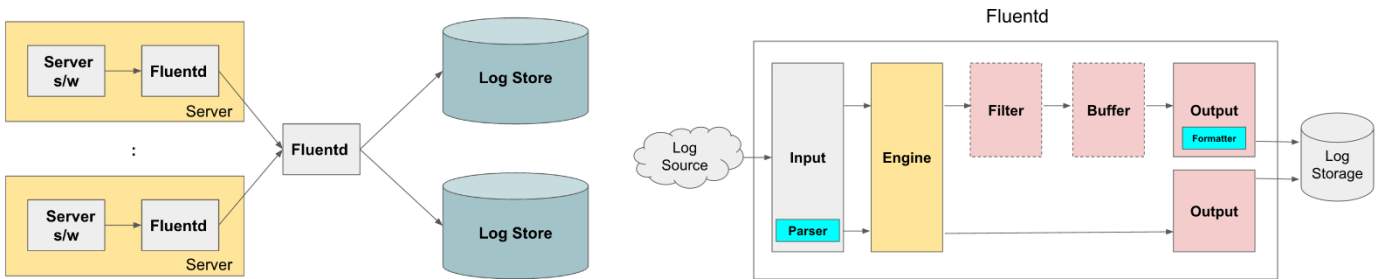
스크라이브는 신뢰성을 염두에 두면서도 무거운 프로토콜과 막대한 디스크 용량을 요구하지 않도록 설계되었다. 스크라이브는 간헐적인 연결 노드 실패를 관리하기 위해 모든 노드의 디스크에 데이터를 스푼링하지만 모든 메시지에 대해 로그 파일을 동기화하지는 않는다. 이로 인해 재난적인 하드웨어 실패나 충돌 시에 약간의 데이터 손실이 발생할 가능성이 있다. 그러나 이 정도의 신뢰성은 대부분의 페이스북 용례에 적합한 편이다.



### 1-3. 데이터 수집 - Fluentd

**Fluentd** - C와 Ruby로 작성된 Fluentd는 수많은 소스에서 발생하는 다양한 포맷의 데이터를 단일 포맷(Json)으로 리포맷(reformat)하고, 이렇게 하여 수집된 데이터를 다양한 대상으로 라우팅(routing) 할 수 있는 로그 수집기(Log Aggregator)

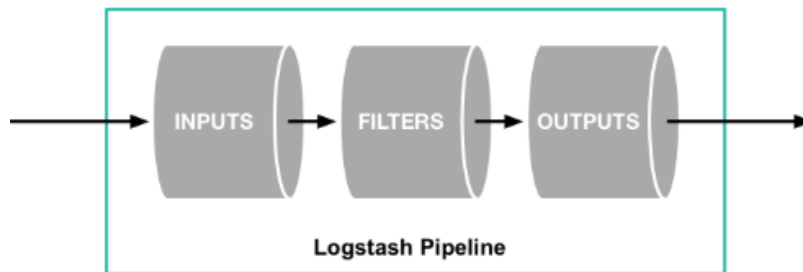
Fluentd는 Input, Parser, Engine, Filter, Buffer, Ouput, Formatter의 총 7개 컴포넌트로 구성되어 있으며, 각 컴포넌트마다 제공되는 플러그인(Plug-in)을 적절하게 활용하고 사용자 환경에 최적화하여 데이터를 수집, 파싱(parsing) 하고 저장 가능



## 1-4. 데이터 수집 - Logstash

**Logstash** - Logstash는 실시간 파이프라인 기능을 가진 오픈소스 데이터 수집 엔진으로 서로 다른 소스의 데이터를 탄력적으로 통합하고 사용자가 선택한 목적지로 데이터를 정규화할 수 있음  
다양한 고급 다운스트림 분석 및 시각화 활용 사례를 위해 모든 데이터를 정리하고 대중화가 가능

Fluentd는 Input, Parser, Engine, Filter, Buffer, Output, Formatter의 총 7개 컴포넌트로 구성되어 있으며, 각 컴포넌트마다 제공되는 플러그인(Plug-in)을 적절하게 활용하고 사용자 환경에 최적화하여 데이터를 수집, 파싱(parsing) 하고 저장이 가능



ELK는 위 그림과 같이, 분석 및 저장 기능을 담당하는 **ElasticSearch**, 수집 기능을 하는 **Logstash**, 이를 시각화하는 도구인 **Kibana**의 앞글자만 딴 단어이다. ELK는 접근성과 용이성이 좋아 최근 가장 핫한 Log 및 데이터 분석 도구

### 1) ElasticSearch

- ElasticSearch는 Lucene 기반으로 개발한 분산 검색엔진으로, Logstash를 통해 수신된 데이터를 저장소에 저장하는 역할을 담당
- 데이터를 중심부에 저장하여 예상되는 항목을 검색하고 예상치 못한 항목을 밝혀낼 수 있다.
- 정형, 비정형, 위치정보, 메트릭 등 원하는 방법으로 다양한 유형의 검색을 수행하고 결합할 수 있다.

### 2) Logstash

- 오픈소스 서버측 데이터 처리 파이프라인으로, 다양한 소스에서 동시에 데이터를 수집하고 변환하여 stash 보관소로 보낸다.
- 수집할 로그를 선정해서, 지정된 대상 서버(ElasticSearch)에 인덱싱하여 전송하는 역할을 담당하는 소프트웨어

### 3) Kibana

- 데이터를 시각적으로 탐색하고 실시간으로 분석 할 수 있다.
- 시각화를 담당하는 HTML + Javascript 엔진이라고 보면 된다.



## 1-4. 데이터 수집 - Fluentd vs Logstash

LogStash와 Fluentd 모두 Windows와 Linux 등 Ruby가 돌아가는 환경에서 모두 동작한다. 명확하게 말하자면, LogStash는 JRuby(Java 필요)이며, Fluentd는 CRuby 기반이다. 결합된 형태로 주로 동작하기 때문에 해당 플랫폼을 위한 플러그인이 지원하는지 여부를 먼저 확인할 필요가 있다.

LogStash는 20개의 Fixed-Size Event를 제한된 On-Memory queue에 담기 때문에, 재시작시 지속성을 위해 External queue의 의존도를 높인다. 이는 LogStash의 잘 알려진 문제로 Redis나 Kafka를 버퍼로 사용함으로써 문제를 해결할 수 있다. 단점에 대한 해결-접근성은 좋은편이지만, 안정성을 위해 독립적으로 동작시키기 어렵다는건 분명한 단점이다. Fluentd는 in-memory 또는 디스크를 활용할 수 있는 고도화된 버퍼링 시스템을 지원한다. 물론, 매개변수를 별도로 구성해야 한다는 단점이 있지만, 2개의 미들웨어를 이해해야하는 LogStash 보단 낫다

	Platform	Event Routing	Plugin Ecosystem	Transport	Performance
<b>Logstash</b>	Mac & Windows	Algorithmic statements	Centralized	Deploy with Redis for reliability.	Uses more memory. Use Elastic Beats for leafs.
<b>Fluentd</b>	Mac & Windows	Tags	Decentralized	Built-in reliability but hard to configure.	Uses less memory. Use Fluent Bit and Fluentd Forwarder for leafs.

<https://grip.news/archives/1340>

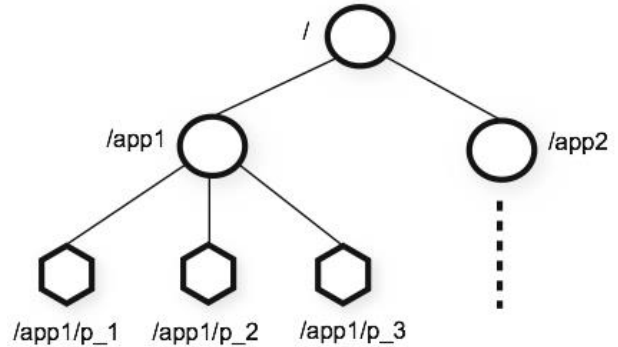
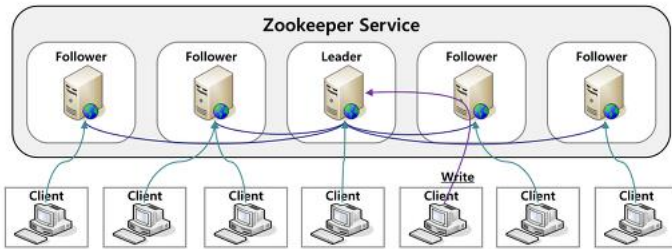


## 분산 코디네이터 - Zookeeper

분산 환경에서 서버들간에 상호 조정이 필요한 다양한 서비스를 제공하는 시스템

첫째, 하나의 서버에만 서비스가 집중되지 않도록, 서비스를 알맞게 분산하여 동시에 처리  
둘째, 하나의 서버에서 처리한 결과를 다른 서버들과도 동기화하여 데이터의 안정성을 보장  
셋째, 운영(active) 서버가 문제가 발생해서 서비스를 제공할 수 없을 경우, 다른 대기 중인 서버를 운영서버로 바꿔서 서비스가 중지 없이 제공.

넷째, 분산 환경을 구성하는 서버들의 환경설정을 통합적으로 관리.



계층형 구조

Kafka에서도 metadata 부분을 zookeeper에 기록하는 식으로 구성함  
분산된 노드에서 공유된 파일시스템이 필요하다면 zookeeper 고려  
데이터를 메모리에 올려놓음으로써 높은 처리량과 낮은 지연율이 장점



## 워크플로우 관리(Workflow Scheduler) – Oozie

하둡 작업을 관리하는 워크플로우 및 코디네이터 시스템으로써 자바 서블릿 컨테이너에서 실행되는 자바 웹 애플리케이션 서버  
Mapreduce, Pig작업 같은 특화된 액션들로 구성된 워크 플로우를 제어함

### Scheduling

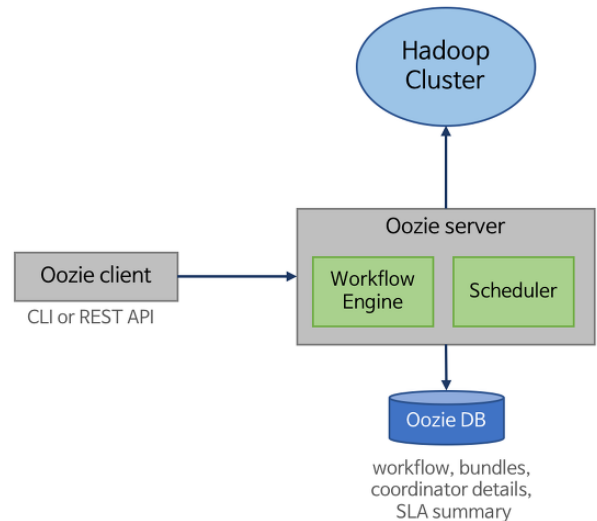
- 특정시간 액션 수행
- 주기적 간격 이후 액션 수행
- 이벤트 발생시 액션 수행

### Coordinating

- 이전 액션이 성공적으로 끝나면 다음 액션 시작

### Managing

- 액션 성공 및 실패 이메일 알림
- 액션수행 시간 및 액션 단계 저장



### 구성 요소

#### Workflow Engine

- 워크플로우를 실행
- 하나의 워크플로우는 여러개의 액션을 포함

#### Coordinator(Scheduler)

- 미리 지정된 위치의 데이터셋의 존재 여부나 frequency에 따라 워크플로우를 스케줄링

#### REST API

- 실행, 스케줄, 워크플로우 모니터링하는 API가 있음

#### CLI

- 커맨드라인을 통하여 작업을 실행하거나 스케줄링, 모니터링 가능

#### Bundle

- 코디네이터를 모아서 한번에 제어하게 해주는 단위

#### Notifications

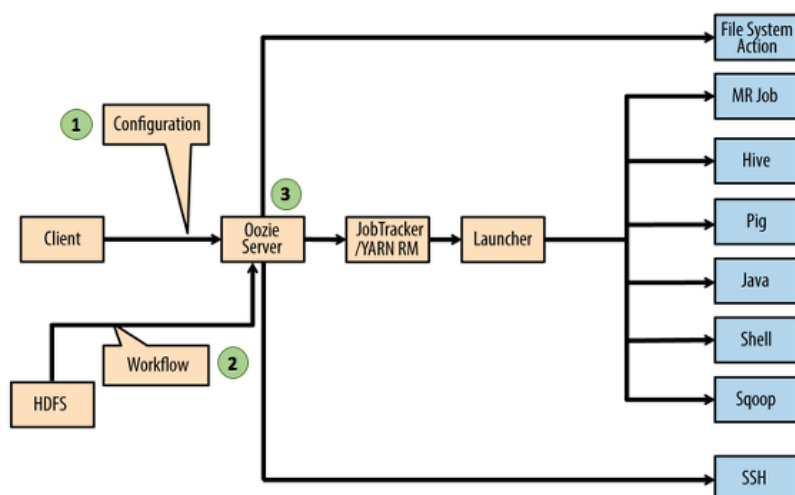
- 작업 상태가 변경 여부에 따라 이벤트를 보내줌
- SLA(Service Level Agreement)

#### monitoring

- 시작, 종료 시간이나 지속 시간을 기반으로 하여 작업에 대한 SLA를 추적하는데 어떤 작업이 SLA를 달성하거나 못하는지 체크하여 사용자에게 통지해줌

#### Database

- 코디네이터, 번들 SLA 및 workflow 이력 등을 저장














## 워크플로우 관리(Workflow management) – Ambari

암바리(Ambari)는 하둡 클러스터에서 각 시스템 리소스를 관리하고 모니터링하는 운영 프레임 워크(Framework)

시스템 관리자는 Ambari를 사용하여 Hadoop 클러스터 를 프로비저닝, 관리 및 모니터링하고 Hadoop을 기존 엔터프라이즈 인프라와 통합이 가능

하둡시스템의 여러 머신의 솔루션을 설치 - 관리하는 부분에서 직접 ssh 접속, 설정하는 것은 매우 어려움

암바리는 클러스터상에 설치된 여러가지 솔루션의 설정값을 관리하고, 각 요소들을 중지, 시작 하는 것을 웹 인터페이스를 통해 쉽게 조작이 가능하다.

<div> Ambari</div> <div>hadoop 0 ops 0 alerts</div> <div>Dashboard Services Hosts Alerts Admin</div> <div>admin</div>				
Actions Groups: All (44)				
Alert Definition Name	Status	Service	Last Status Changed	State
Any	All	All	Any	All
 Metric Monitor Status	OK (5)	AMS	18 hours ago	Enabled
 Ambari Agent Disk Usage	OK (5)	Ambari	about a day ago	Enabled
 DataNode Web UI	OK (4)	HDFS	18 hours ago	Enabled
 DataNode Process	OK (4)	HDFS	18 hours ago	Enabled
 DataNode Storage	OK (4)	HDFS	2 hours ago	Enabled
 NodeManager Health	OK (4)	YARN	18 hours ago	Enabled
 NodeManager Web UI	OK (4)	YARN	18 hours ago	Enabled
 Metric Collector HBase Maser CPU Utilization	OK	AMS	about a day ago	Enabled
 Metric Collector - HBase Master Process	OK	AMS	18 hours ago	Enabled
 Metric Collector Process	OK	AMS	18 hours ago	Enabled

## 워크플로우 관리(Workflow management) – Cloudera

[https://www.zdnet.co.kr/view/?no=20130503081937&re=R\\_20130529081754](https://www.zdnet.co.kr/view/?no=20130503081937&re=R_20130529081754)

## 워크플로우 관리(Workflow management) – Cloumon

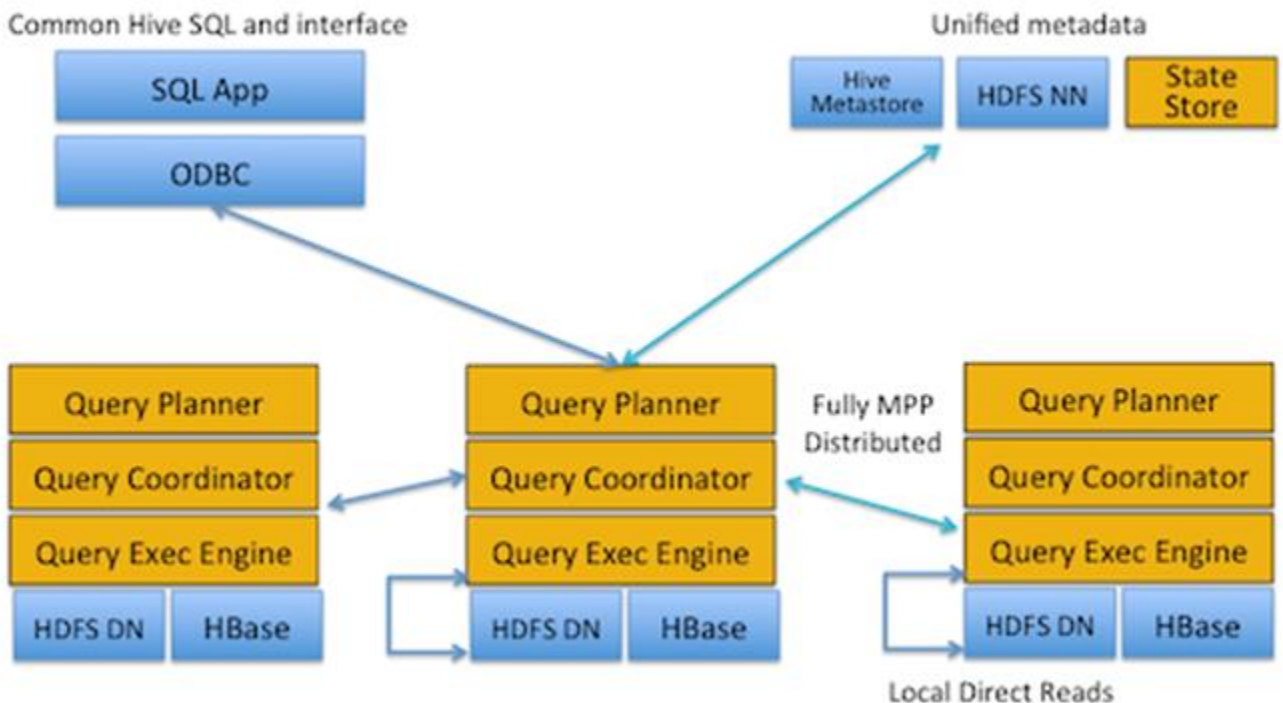
[https://www.zdnet.co.kr/view/?no=20130503081937&re=R\\_20130529081754](https://www.zdnet.co.kr/view/?no=20130503081937&re=R_20130529081754)

## 실시간 SQL질의(Real-time in Data Analytics) - Impala(feat.Cloudera)

Impala는 HDFS에 저장돼 있는 데이터를 SQL을 이용해 실시간으로 분석할 수 있는 시스템  
MapReduce 프레임워크를 이용하지 않고 분산 질의 엔진을 이용하여 분석하여 빠른 결과를 제공

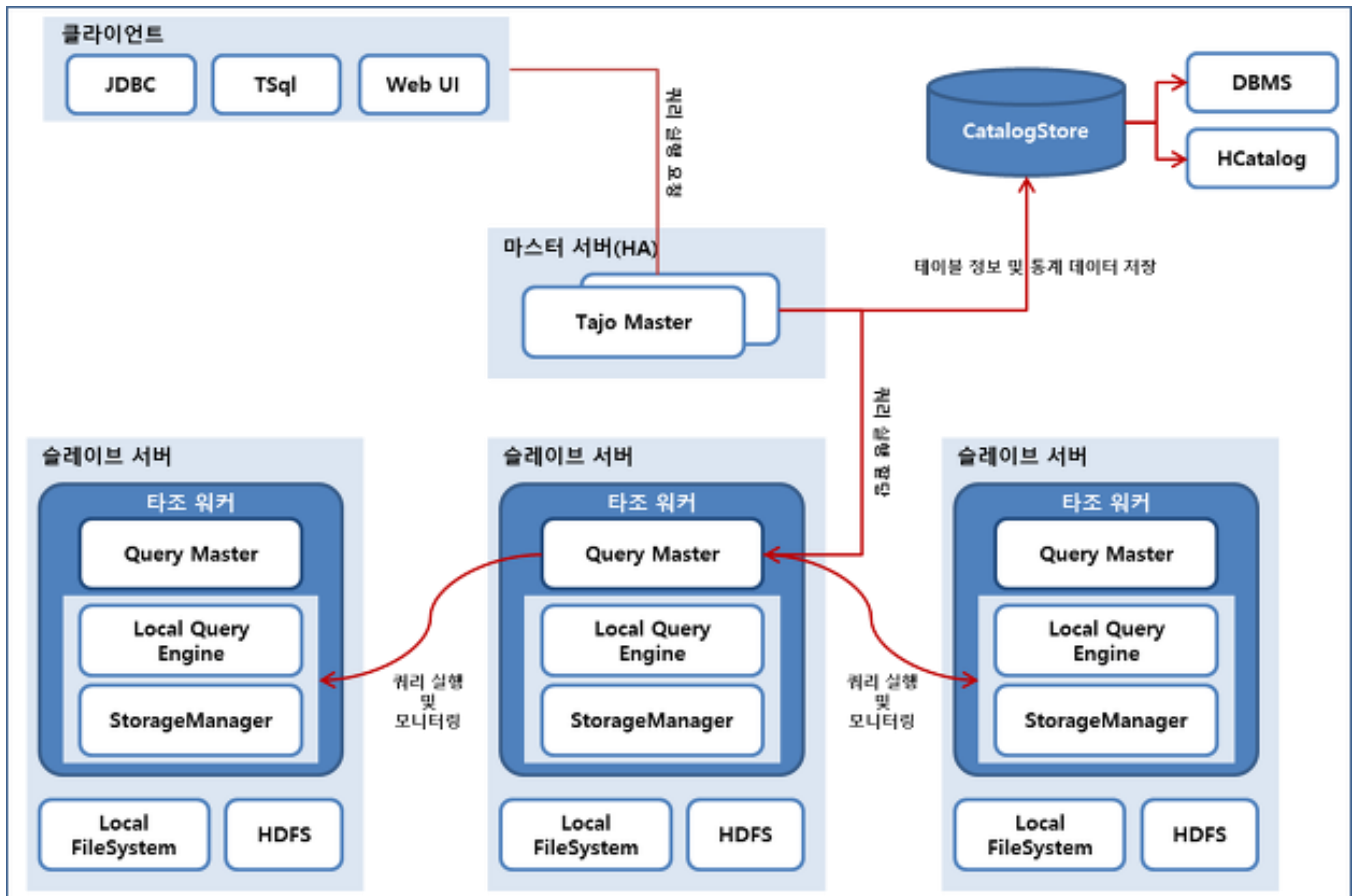
Impala와 Hive의 차이는 실시간성 여부이며, Hive는 데이터 접근을 위해 MapReduce프레임워크를 이용하지만, Impala는 응답시간을 최소한으로 줄이기 위해 고유의 분산 질의 엔진을 사용, 이 분산 질의 엔진은 클러스터 내 모든 데이터 노드에 설치

Impala는 Hive보다 CPU 부하를 줄였고, 줄인 만큼 I/O 대역폭을 이용할 수 있다. 그래서 순수 I/O bound 질의의 경우 Impala는 Hive보다 3~4배 좋은 성능 결과를 보여준다.  
질의가 복잡해지면 Hive는 여러 단계의 MapReduce 작업 또는, Reduce-side 조인(JOIN) 작업이 필요하다. 이처럼 MapReduce 프레임워크로 처리하기에 비효율적인 질의(적어도 하나 이상의 JOIN 연산이 들어간 질의)의 경우 Impala가 7~45배 정도 더 좋은 성능을 보인다.  
분석할 데이터블록이 파일 캐시되어 있는 상태라면 매우 빠른 성능을 보여주고, 이 경우 Hive보다 20~90배 빠른 성능을 보여준다.



## 실시간 SQL질의(Real-time in Data Analytics) - Tajo(feat.고려대학교)

Tajo는 하둡기반의 대용량 데이터웨어 하우스 시스템으로 SQL표준을 지원,  
질의 전체를 분산처리하며 HDFS를 기본저장소로 사용하여 질의 실행 결과가 HDFS에 저장  
롱타임 질의에 해당하는 ETL작업 뿐만 아니라 로우 레이턴시 질의도 지원  
100밀리세컨드부터 수시간까지 실행되는 질의를 처리 할 수 있음  
사용자가 직접 함수를 정의하며  
다양한 최적화를 위해 비용기반최적화 모델(Cost based optimization model)과 확장 가  
능한 리라이트 룰(Rewrite Rule)을 제공



## 데이터 분석 - Hive, Pig

Apache Pig와 Hive는 Hadoop 위에 레이어 된 두 개의 프로젝트이며 Hadoop의 MapReduce 라이브러리를 사용하기위한 고급 언어를 제공합니다.

Apache Pig는 MapReduce가 원래 설계 한 작업과 마찬가지로 데이터 읽기, 필터링, 변형, 결합 및 작성과 같은 작업을 설명하는 스크립팅 언어를 제공합니다. Pig는 MapReduce를 직접 사용하는 수천 줄의 Java 코드에서 이러한 작업을 표현하는 대신 사용자가 bash 또는 perl 스크립트와 달리 언어로 표현할 수 있게합니다. Pig는 Java 자체로 MapReduce 작업을 코딩하는 것과 달리 MapReduce 기반 작업을 프로토 타이핑하고 신속하게 개발하는 데 탁월합니다.

Pig가 “Hadoop 용 스크립팅”인 경우 Hive는 “Hadoop 용 SQL 쿼리”입니다. Apache Hive는 Hadoop에서 여러 MapReduce 작업의 단계별 스크립트를 직접 스크립팅하는 대신 Hadoop 작업을 실행하여 데이터를 쿼리하기 위해보다 구체적이고 고급 수준의 언어를 제공합니다. 언어는 설계 상 매우 SQL과 유사합니다.

Hive는 대용량 데이터에 대해 장기 실행 일괄 쿼리를 수행하는 도구로 사용됩니다. 그것은 어떤 의미에서는 “실시간”이 아닙니다. 하이브는 SQL과 같은 쿼리 및 비즈니스 인텔리전스 시스템에 익숙한 분석가 및 비즈니스 개발 유형을위한 훌륭한 도구입니다. 반짝이는 새 Hadoop 클러스터를 사용하여 위에서 언급 한 스토리지 시스템에 저장된 데이터를 통해 임시 쿼리를 수행하거나 보고서 데이터를 생성 할 수 있습니다.



# Hadoop echo system

