

## Part III : 수집/저장/처리 기술

### 3.1 수집/저장/처리 기술 개요



# 1. 개요



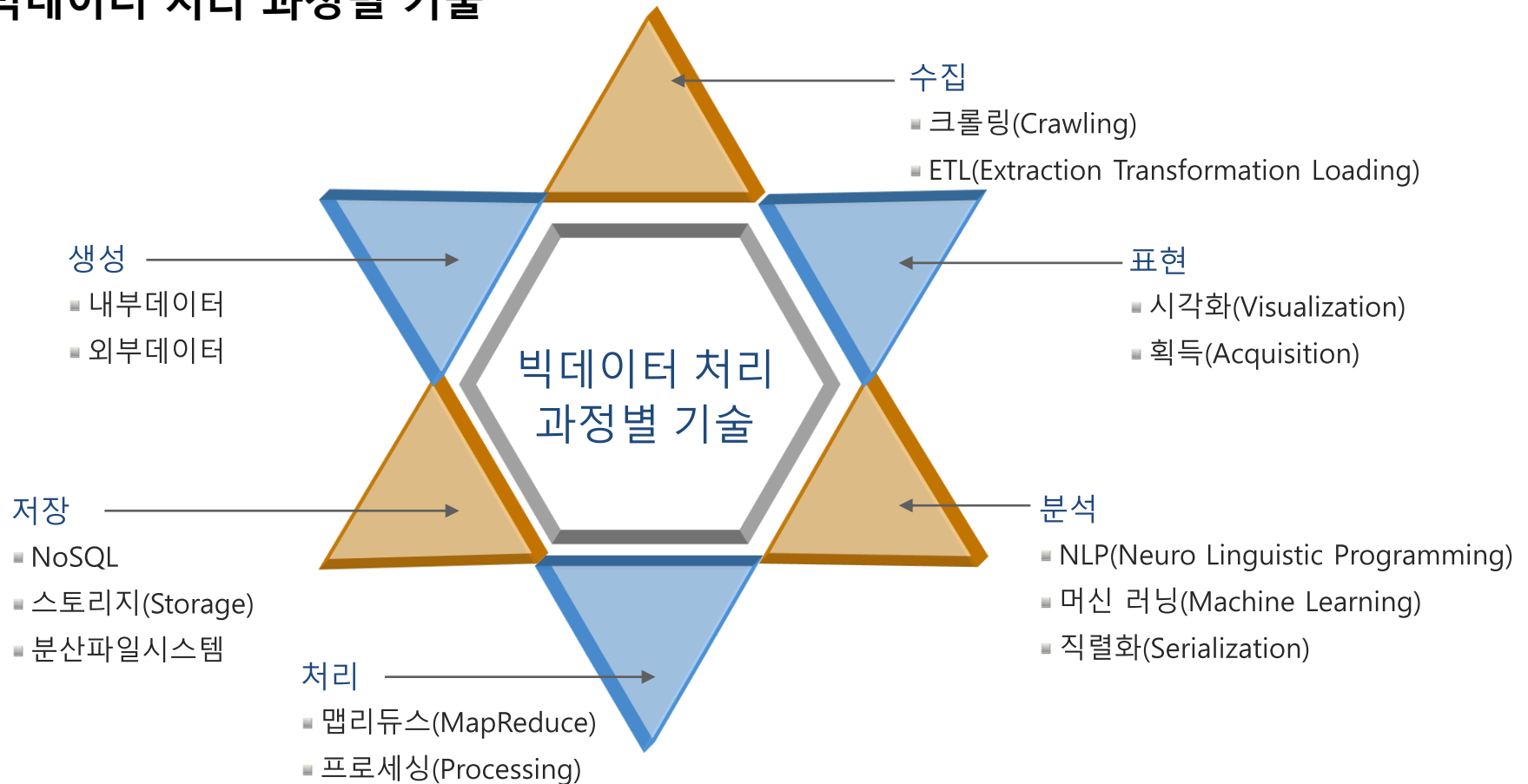
## ● 빅데이터 처리 과정



# 1. 개요



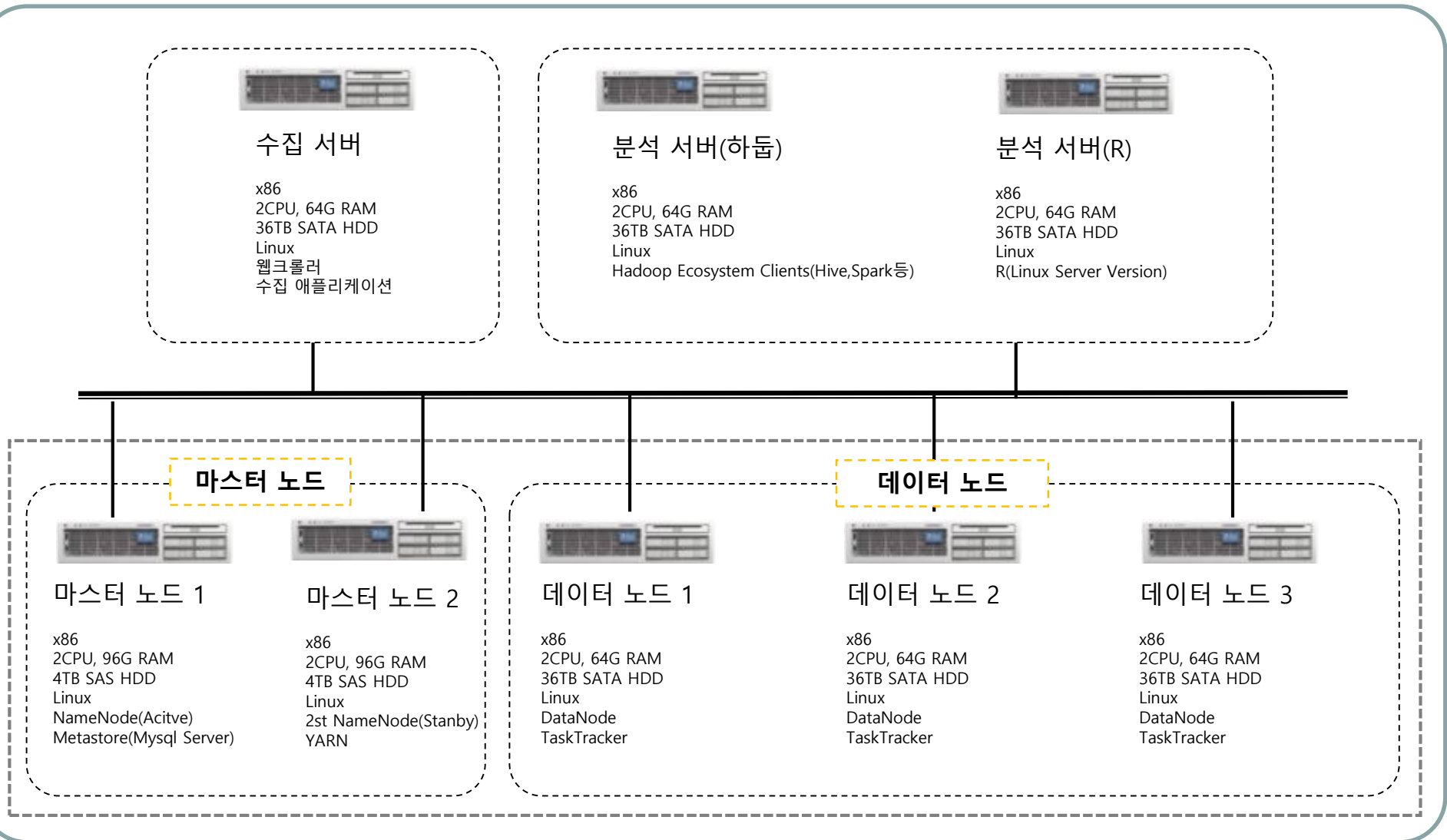
## ● 빅데이터 처리 과정별 기술



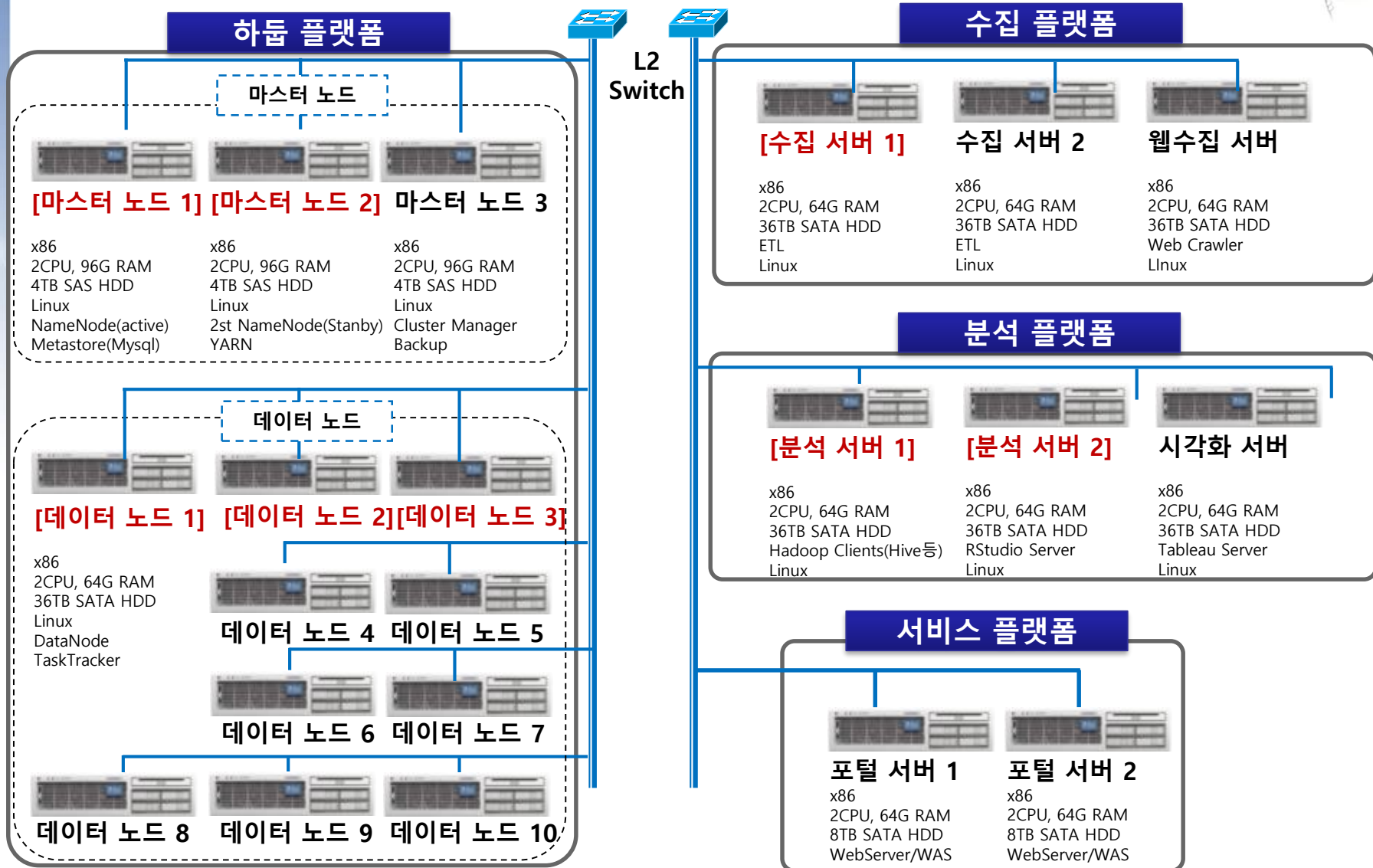
## 2. 목표시스템 인프라 구성도



### 개발용 최소 구성



## 2. 목표시스템 인프라 구성도



## 2. 목표시스템 인프라 구성도



### To-Be 인프라 H/W 사양

마스터 노드 : 2대	
TYPE	x86
CPU	2 × 6 core 2.9 Ghz/15 MB cache
메모리	96 GB DDR3-1600 ECC
디스크 컨트롤러	SAS 6 Gb/s
디스크	4 × 1 TB SAS 15000 RPM ( RAID )
NIC	2 × 1 Gb Ethernet
비고	Hadoop & Ecosystem Linux

데이터 노드 : 3대	
TYPE	x86
CPU	2 × 6 core 2.9 Ghz/15 MB cache
메모리	64 GB DDR3-1600 ECC
디스크 컨트롤러	SAS 6 Gb/s
디스크	12 × 3 TB SATA HDD ( JBOD )
NIC	2 × 1 Gb Ethernet
비고	Hadoop & Ecosystem Linux

상용분석/DB 서버 : 1대	
TYPE	UNIX
CPU	2 × 6 core 2.9 Ghz/15 MB cache
메모리	96 GB DDR3-1600 ECC
디스크 컨트롤러	SAS 6 Gb/s
디스크	4 × 1 TB SAS 15000 RPM ( RAID )
NIC	2 × 1 Gb Ethernet
비고	SAS RDBMS(Oracle)

수집/검색 서버 : 1대    분석서버:1대	
TYPE	x86
CPU	2 × 6 core 2.9 Ghz/15 MB cache
메모리	64 GB DDR3-1600 ECC
디스크 컨트롤러	SAS 6 Gb/s
디스크	12 × 3 TB SATA HDD ( JBOD )
NIC	2 × 1 Gb Ethernet
비고	- 웹크롤러, Hadoop Ecosystem Client - R Server & Hadoop Ecosystem Client Linux

### 3. 소프트웨어 구성도



#### To-Be 신규 구현 대상 시스템 응용기능 체계도

##### # 주요 오픈소스 S/W

구분	설명	관련 S/W
원본 데이터 저장	대용량 분산 파일 시스템	하둡 HDFS
구조적 데이터 저장	대용량 분산 데이터 저장소	NoSQL-HBase, Hive, 카산드라, 몽고DB, Couch DB
배치 분산병렬 처리	병렬처리 시스템/프레임워크	하둡 MapReduce, Pig, HiveQL
스트리밍 프로세싱	실시간 스트리밍 처리 시스템	S4, Storm
데이터 마이닝	기계학습	Mahout
데이터 수집	데이터 수집, RDBMS와의 데이터 교환	Flume, Sqoop, Scribe, Chukwa
통계 및 분석	통계분석 및 시각화	R
클러스터 관리	하둡 클러스터 관리 및 모니터링	Cloudera Cluster Manager, Hortonworks Ambari, Ganglia
데이터 직렬화	이기종간 데이터 전송/직렬화	Thrift, Avro, RHadoop

##### # 적용 S/W

S/W	설명	비고
하둡 HDFS	하둡분산파일시스템, 대용량 데이터를 다수의 데이터노드에 분산 저장, 데이터가용성 보장	오픈소스
NoSQL-HBase	HDFS 기반의 Semi-Structured 데이터 저장소, 정형 및 비정형 대용량 데이터 저장	오픈소스
하둡 MapReduce	HDFS 기반의 대용량 데이터 배치 병렬처리 시스템 및 프레임워크(개발환경)	오픈소스
Hive	하둡 기반의 DataWare House, HiveQL이라는 쿼리 지원, HDFS에 대용량데이터 자동병렬처리	오픈소스
Pig	Pig Latin 스크립트를 작성하면 HDFS에 저장된 데이터로 맵리듀스 실행	오픈소스
R(Linux Server Version)	오픈소스 통계 분석 및 시각화 소프트웨어, 리눅스 서버 버전을 지원	오픈소스, 상용(RA)
RHadoop	HDFS나 HBase의 데이터를 R에서 사용하도록 지원하는 커넥터, RMR은 R 함수 맵리듀스 처리	오픈소스
Sqoop	하둡(HDFS,Hive,Hbase)와 RDBMS(Oracle,Mysql)간의 데이터 Import/Export 지원 어플리케이션	오픈소스
Zookeeper	하둡기반의 클러스터 분산코디네이터, HBase의 마스터 데이터 저장 및 리전 서버 관리 지원	오픈소스,
Cluster Manager	하둡과 수 많은 에코시스템 S/W를 여러 대의 머신에 설치, 관리, 설정동기화, 모니터링	Ambari(오픈소스) CM(클라우드라,상용)
웹크롤러	웹문서(게시판,카페 등)를 수집해서 저장하는 어플리케이션	개발 또는 상용솔루션

## Part III : 수집/저장/처리 기술

### 3.2 빅데이터 수집 기술

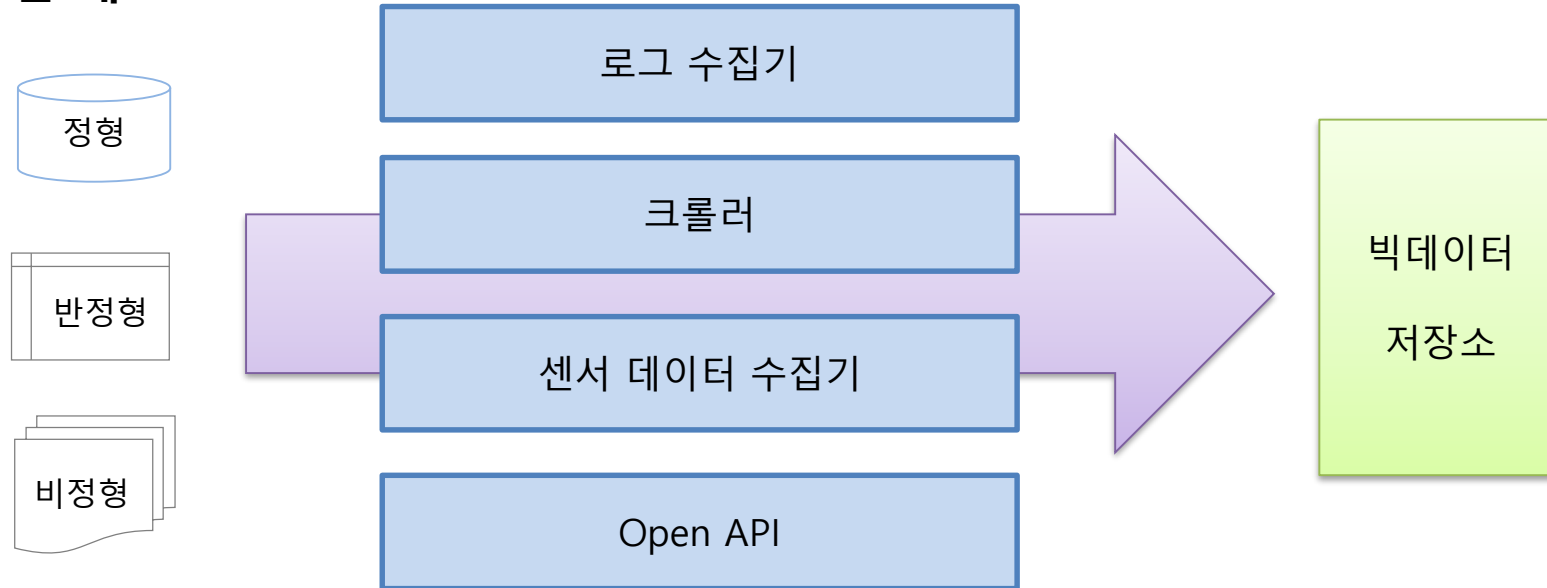




# 1. 빅데이터 수집 기술



## ● 수집 개요



## <빅데이터의 주요 수집 기술>

기술	개발	최초 공개	주요 기능 및 특징
Sqoop	아파치	2009년	RDBMS와 HDFS(NoSQL) 간의 데이터 연동
Flume	Cloudera	2010년	방대한 양의 이벤트 로그 수집
Kafka	Linkedin	2010년	분산 시스템에서 메시지 전송 및 수집

## 2. SQOOP



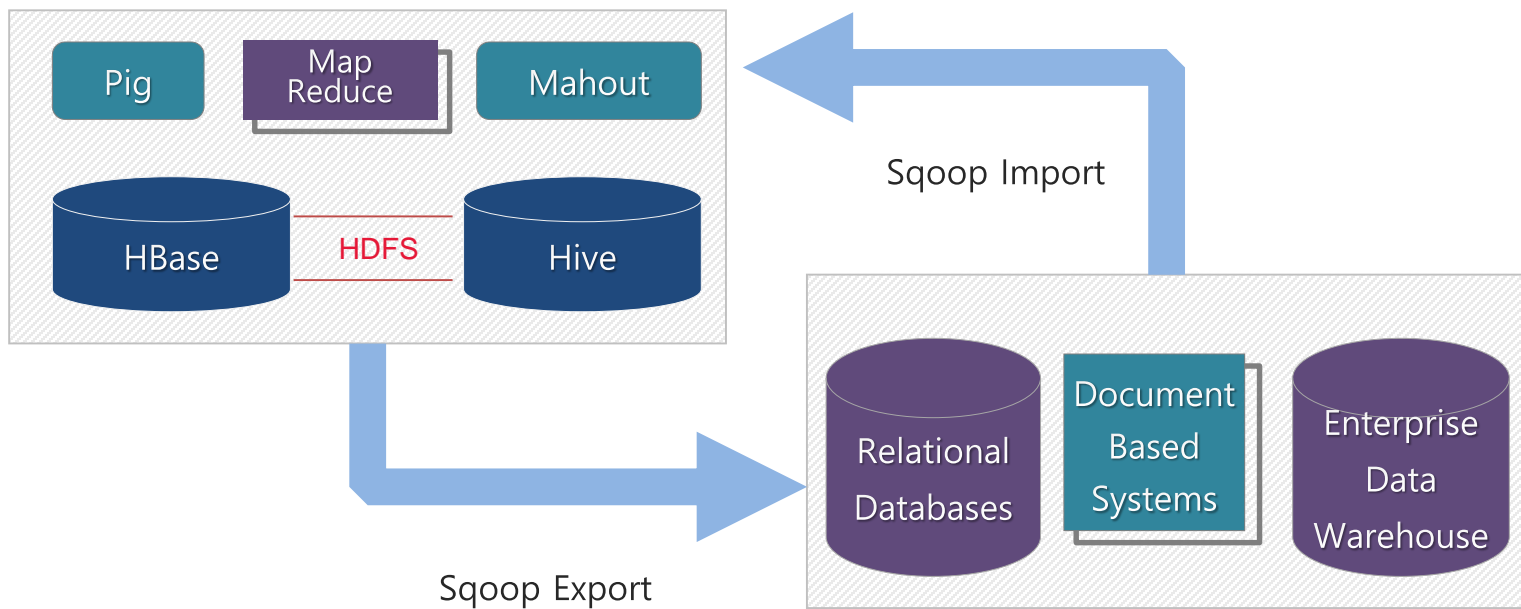
### ● SQOOP

#### ■ 개요

- 하둡과 데이터베이스 간 데이터 이동
- 양방향 : Import & Export
- 다양한 DB로부터의 자료 이동 지원

#### ■ 특징

- JDBC드라이버를 통한 연동 : RDBMS(MySQL, PostgreSQL, 오라클)와 NoSQL(몽고DB, Redis)과 같은 다양한 데이터베이스 지원
- 하둡 에코시스템 (Hive, pig, HBase등)와 연계

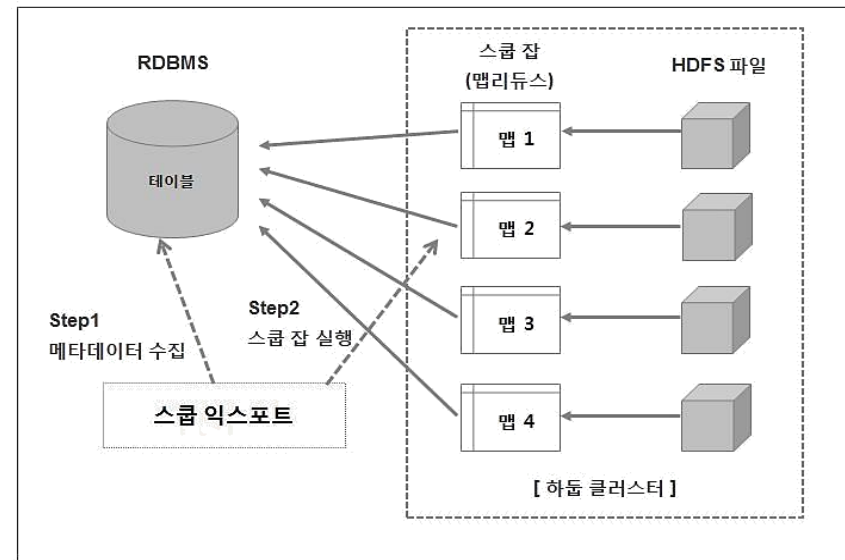
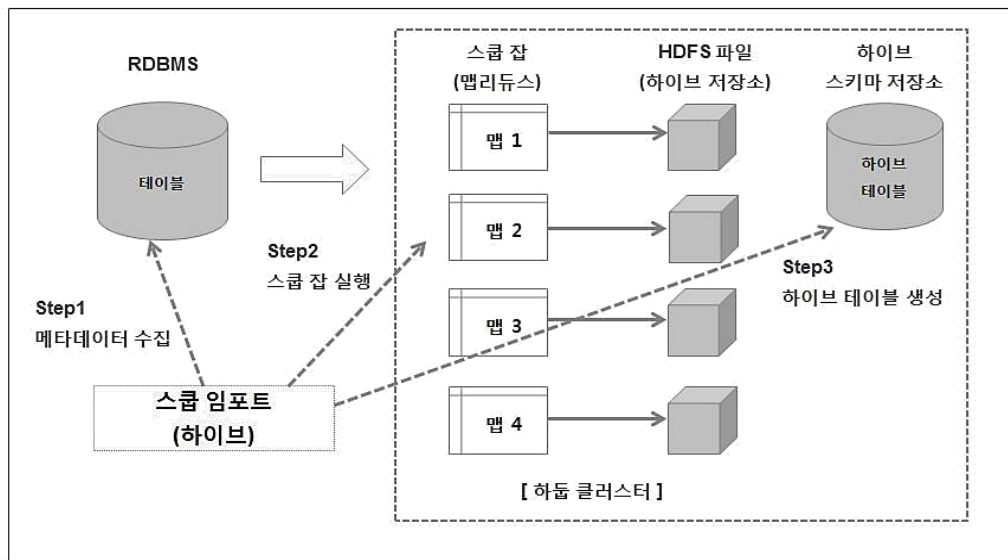
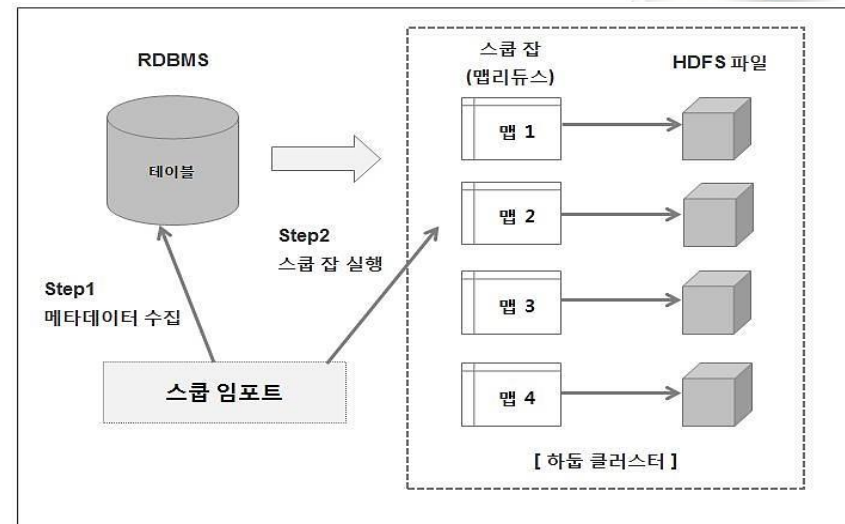


## 2. SQOOP



### ● SQOOP

- 스쿱 임포트(Import)
  - RDBMS -> HDFS
- 하이브 임포트(Import)
  - RDBMS -> HDFS -> Hive Table
- 스쿱 익스포트(Export)
  - HDFS -> RDBMS



# 3. Flume



## ● Flume

### ■ 개요

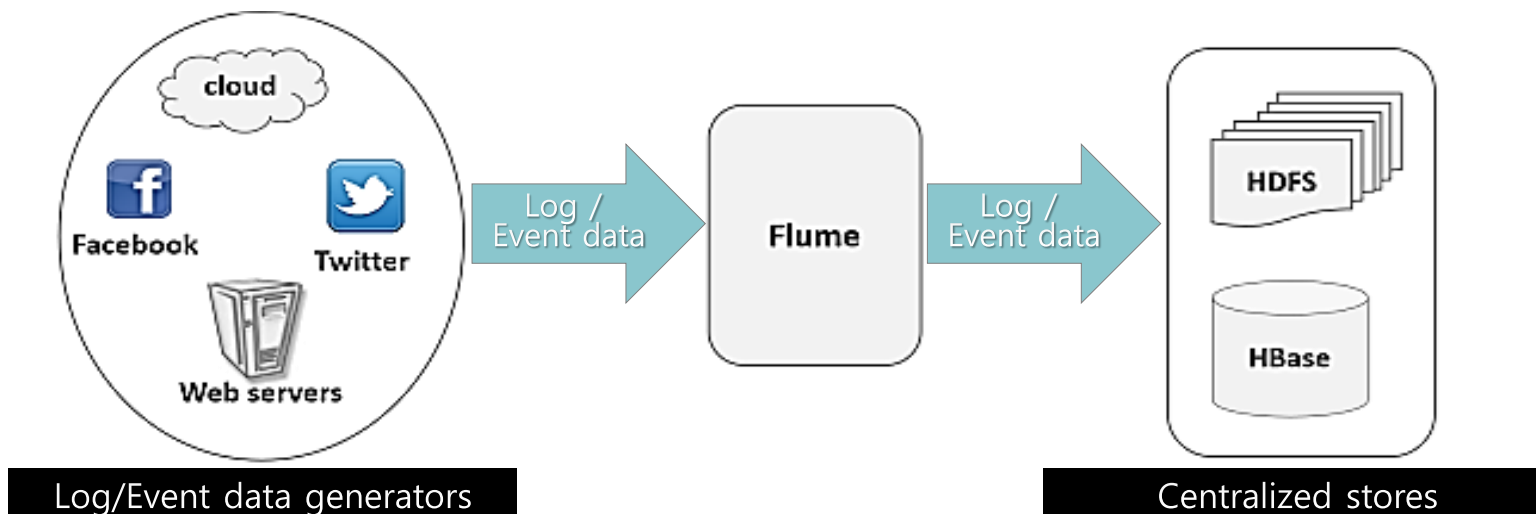
- 오픈 소스 로그 수집 소프트웨어
- 분산 환경에서 대량의 로그 데이터를 효과적으로 수집 후 전송

### ■ 특징

- 다양한 로그 데이터 수집 및 모니터링이 가능하며 실시간 전송을 지원
- 자바로 구현되어 있기 때문에 다양한 운영체제에 설치가 가능
- 장애에 쉽게 대처 가능

### ■ 배경 및 필요성

- 기존의 다양한 시스템들과 하둡 사이에 연동하는 방법 개선 필요



# 3. Flume



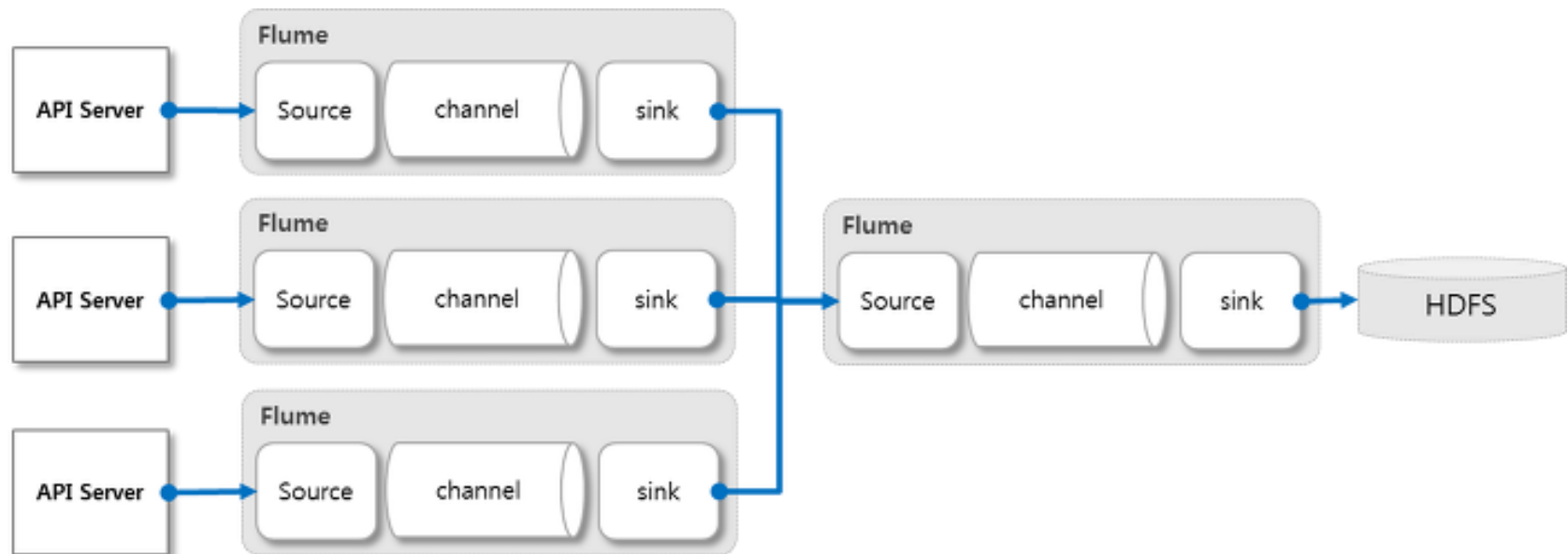
## ● Flume

### ■ 아키텍처

- 에이전트 : 소스, 채널, 싱크로 구성
- 개별 에이전트의 연결이 가능

### ■ 구성요소

- 소스 : 지정한 소스를 통해서 이벤트를 받아 채널로 전달한다.
- 채널 : 일종의 큐로서 이벤트를 임시로 보관한다.
- 싱크 : 채널에서 이벤트를 읽어와 출력 대상에 쓴다.



# 4. Kafka



## ● Kafka

### - Kafka 개요

#### ■ 개요

- 확장성이 좋고 처리량이 높은 분산 메시지 시스템  
-> 우수한 메시지 전달 성능을 보장

#### ■ 링크드인과 Kafka

- 링크드인의 실시간 데이터나 오프라인 데이터를 처리하고 수집
- 높은 처리량을 유지하면서 병렬처리에 초점을 맞춘 구조
- 다량의 클라이언트 생성이 가능한 유연성을 제공하고 분산처리를 지원

### - Kafka 아키텍처

#### ■ Kafka의 구성

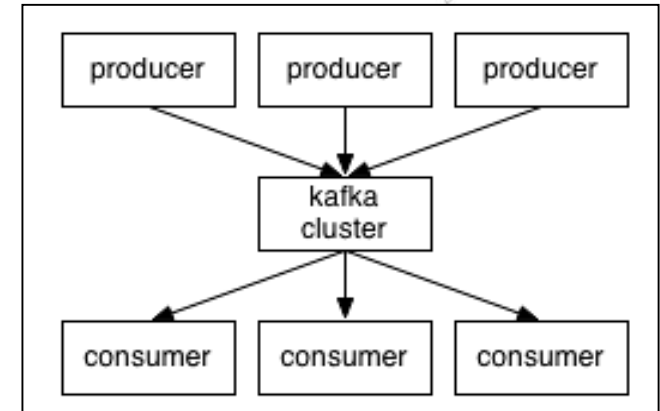
- Producer, Consumer, Broker

#### ■ 역할 및 처리 과정

- Kafka의 Broker는 Topic을 기준으로 메시지를 관리
- Producer는 특정 Topic의 메시지를 생성한 뒤 해당 메시지를 Broker에 전달
- Topic별로 분류하여 쌓아놓으면, 해당 Topic을 구독하는 Consumer들이 메시지를 가져가서 처리

#### ■ 설계 특성

- 확장성(scale-out)과 고가용성(High Availability)



## Part III : 수집/저장/처리 기술

### 3.3 빅데이터 저장 기술



# 1. 빅데이터 저장 기술



## ● 저장 기술

구분	기술	최초 개발	주요 기능 및 특징
분산파일시스템	HDFS	아파치	대표적인 오픈소스 분산파일시스템
	Hive	페이스북	HDFS기반의 DataWarehouse
	S3	아마존	아마존의 클라우드 기반 분산 스토리지 서비스
NoSQL	HBase	아파치	HDFS기반의 NoSQL
	Cassandra	A. Lakshman	ACID 속성을 유지한 분산 데이터베이스
	몽고DB	10gen	DB의 수평 확장 및 범위 질의 지원, 자체 맵리듀스



## 2. HDFS



### ● HDFS

- HDFS 개요
  - 개발 동기
    - 대용량 파일의 스트리밍 읽기와 쓰기에 뛰어난 성능
    - 전통적인 대용량 SAN, NAS 스토리지를 대체
  - HDFS 개요 및 특징
    - 분산 파일시스템 : 개별 디스크와 머신이 지원하는 용량의 한계를 뛰어 넘을 수 있다.
    - 클러스터의 머신들은 파일시스템을 구성하는 전체 데이터를 나누어 각각 저장
    - 단일 중앙 서버에 저장된 파일시스템 메타데이터는 블록 데이터의 디렉터리 역할을 한다.
- HDFS 아키텍처
  - 데이터노드
    - 블록은 한 파일의 분리된 바이너리 파일 조각
    - HDFS에서 블록 데이터를 저장하고 추출하는 기능을 담당하는 데몬
  - 네임노드
    - 파일시스템 메타데이터를 저장하고 파일시스템의 전체 이미지를 관리
    - 데이터노드들은 하트비트(heartbeat)에 자신의 상태를 실어서 주기적으로 네임노드에 보낸다.
      - > 네임노드는 데이터노드들의 생존 여부와 가용 블록 등 전체적인 상황을 파악 가능

# 3. Hive



- Hive 개요

## What is Hive

Hive는 하둡의 상위에 위치한 Data Warehouse 구성요소로 SQL 쿼리를 MapReduce 작업을 위해 컴파일하고 클러스터에서 실행되게 함

### Hadoop/Hive Architecture



Business Analysts can perform data analysis leveraging Hadoop

Relational DWH with table/partition structures

SQL type queries across TB's of data

# 3. Hive



## ● Hive 개요

### ■ 개요

- SQL을 사용, 데이터 요약, 쿼리 및 분석을 수행, 하둡 기반의 데이터웨어하우스 시스템
- 페이스북 주도로 개발

### ■ 기본적인 작동 원리

- 사용자가 SQL 쿼리를 작성하면 이것을 자동으로 맵리듀스 작업으로 변경해서 클러스터에서 실행

### ■ 기본 구성과 특징

- 실행 부분 : 쿼리->맵리듀스 실행
- 메타데이터 정보 : Mysql과 같은 RDBMS에 저장

### ■ Hive와 RDBMS의 차이점

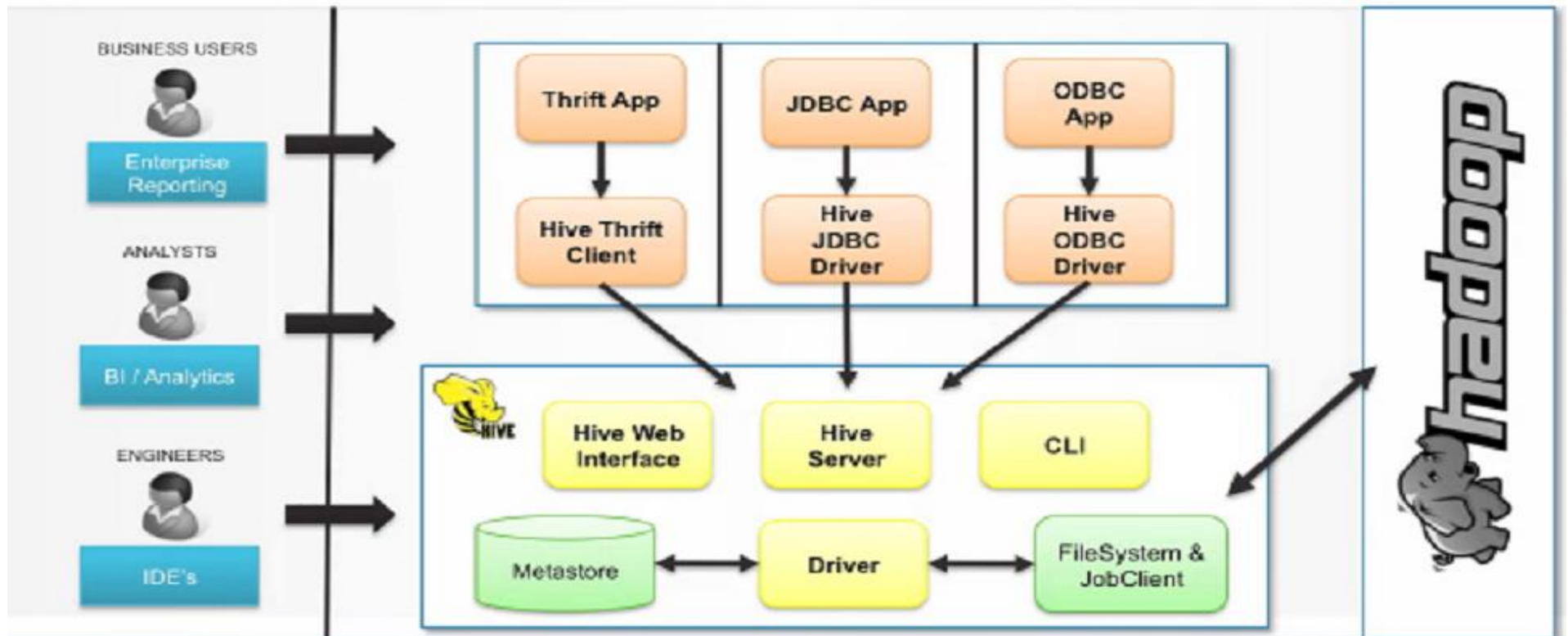
- 작은 데이터일 경우 응답 속도가 느리다.
- 레코드 단위의 Insert, Update, Delete를 지원하지 않는다.
- 트랜잭션을 지원하지 않는다.
- 통계정보를 바로 확인할 수 없다.
- 입력값 오류도 바로 확인할 수 없다.

### 3. Hive



- 엔터프라이즈 Data Warehouse

엔터프라이즈 DW에서의 Hive

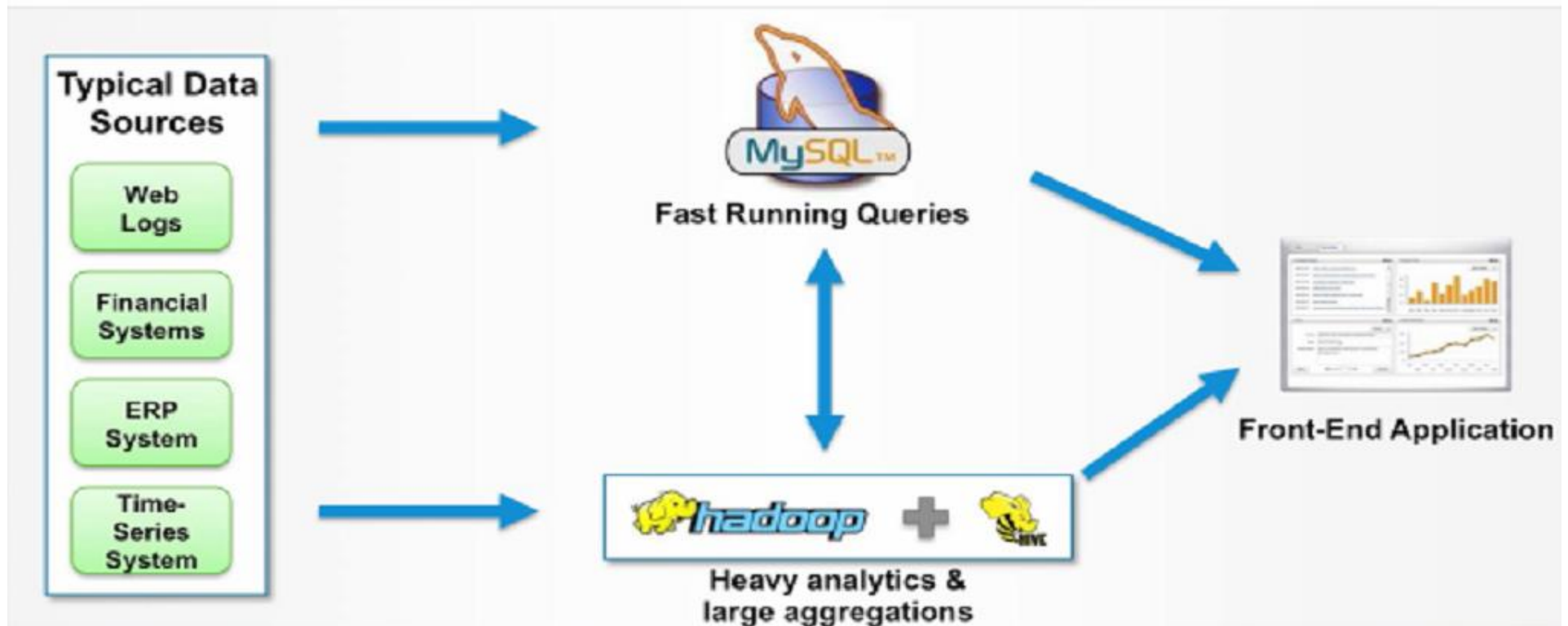


### 3. Hive



- 기존 RDBMS와 Hive의 결합

데이터를 처리하는 새로운 기술적인 방법 제시



# 3. Hive



- **데이터 모델**

- **테이블**

- 컬럼 타입 ( int, float, string, boolean )
    - 리스트 : map ( JSON과 유사 )

- **파티션**

- 테이블은 하나 이상의 파티션 키를 가짐
    - 데이터 접근을 위한 테이블 구조의 최적화(인덱스)  
Ex) 날짜별 파티션 테이블

- **버킷**

- 파티션의 데이터는 특정 해쉬 함수에 의해서 버킷으로 분할
    - 특정 범위의 해쉬 파티션(샘플링에 이용)

# 3. Hive



- **메타스토어(Metastore)**
  - 데이터베이스
    - 테이블 목록등의 정보를 가진 네임스페이스
  - 테이블 명세를 저장
    - 컬럼 타입, 물리적 레이아웃
  - 파티션 데이터
  - Derby, Mysql 또는 RDBMS에 저장될 수 있음



# 3. Hive



- **물리적 레이아웃**

- HDFS에 위치한 Warehouse 디렉토리
  - /user/hive/warehouse
- 테이블은 Warehouse의 서브디렉토리에 저장
  - 파티션은 해당 테이블의 서브디렉토리
- 실제 데이터는 일반파일로 저장됨
  - 특정 구분자로 컬럼을 구분하는 텍스트 파일
  - 또는 순차 파일 형태
  - SerDe 형식, 임의 형식 지원

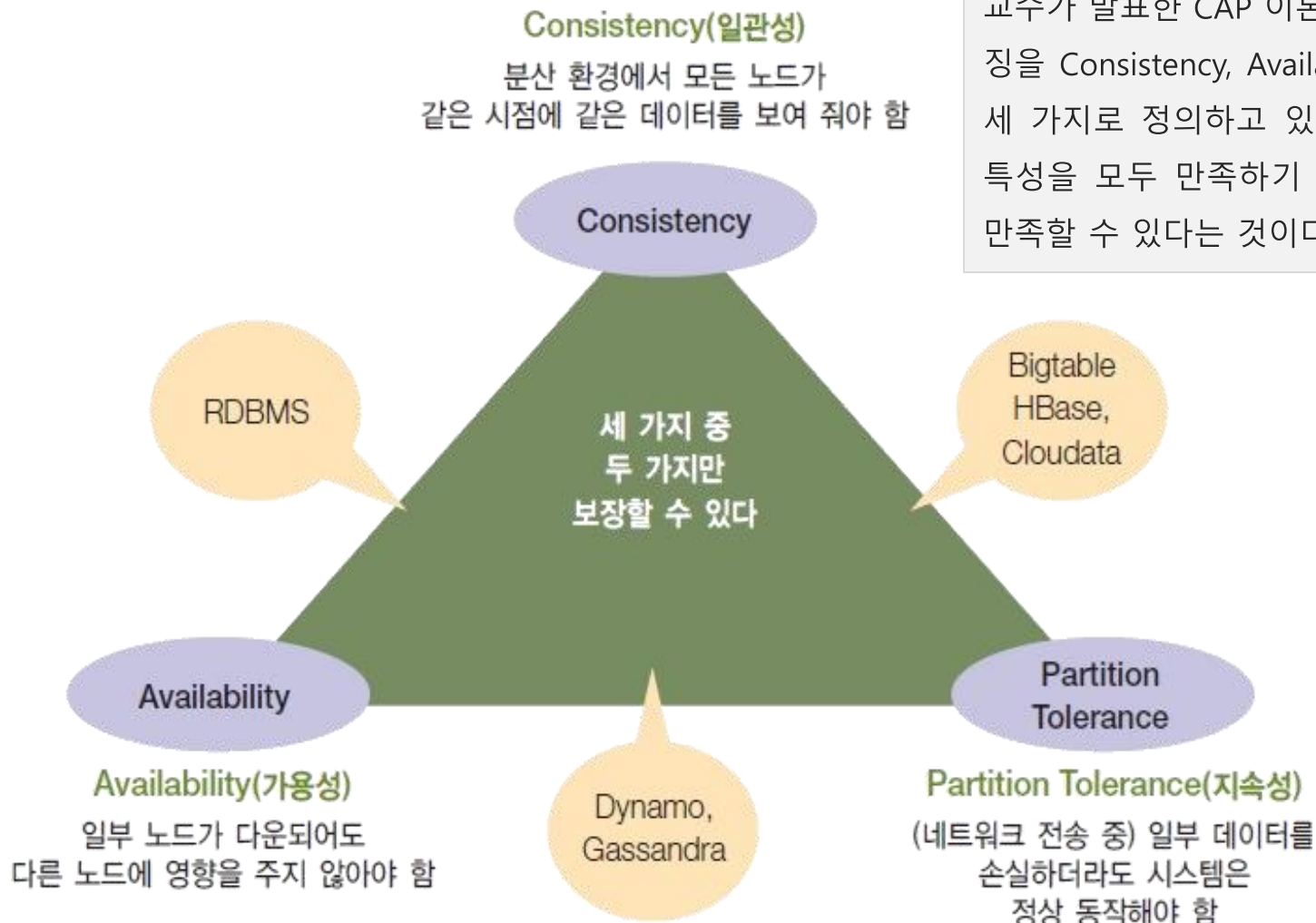


# 4. NoSQL



## ● NoSQL

2002년, 버클리 대학의 에릭 브루어(Eric Brewer) 교수가 발표한 CAP 이론은 분산 컴퓨팅 환경의 특징을 Consistency, Availability, Partition Tolerance 세 가지로 정의하고 있는데, 어떤 시스템이든 이 특성을 모두 만족하기 어려우며 최대 두 가지만 만족할 수 있다는 것이다.



# 4. NoSQL



## ● RDBMS와 NoSQL

- CAP 이론을 기준으로 한 RDBMS와 NoSQL의 비교

구분	설명	적용 예
RDBMS	일관성(C)과 가용성(A)을 선택	트랜잭션 ACID의 보장 (금융 서비스)
NoSQL	일관성(C)과 가용성(A) 중 하나를 포기하고, 지속성(P)를 보장	<ul style="list-style-type: none"><li>• C+P형 : 대용량 분산 파일 시스템(성능 보장)</li><li>• A+P형 : 비동기식 서비스(아마존, 트위터 등)</li></ul>

- RDBMS와 NoSQL의 장 · 단점 및 특성 비교

구분	RDBMS	NoSQL
장·단점	<ul style="list-style-type: none"><li>• 데이터 무결성, 정확성 보장</li><li>• 정규화된 테이블과 소규모 트랜잭션이 있다.</li><li>• 확장성에 한계가 있다.</li><li>• 클라우드 분산 환경에 부적합</li></ul>	<ul style="list-style-type: none"><li>• 데이터의 무결성과 정확성을 보장하지 않는다.</li><li>• 웹 환경의 다양한 정보를 검색 · 저장 가능</li></ul>
특성	UPDATE, DELETE, JOIN 연산 가능 ACID 트랜잭션이 있다. 고정 스키마가 있다.	<ul style="list-style-type: none"><li>• 수정 · 삭제를 사용하지 않는다.(입력으로 대체)</li><li>• 강한 일관성은 불필요</li><li>• 노드의 추가 및 삭제, 데이터 분산에 유연</li></ul>

# 4. NoSQL



## ● NoSQL의 분류와 특성

### NoSQL의 분류

데이터 모델	설명	제품 예
<키,값> 저장 구조	<ul style="list-style-type: none"><li>가장 간단한 데이터 모델</li><li>범위 질의는 사용이 어렵다. (DB에서 지원하면 사용 가능)</li><li>응용 프로그램 모델링이 복잡하다.</li></ul>	아마존 DynamoDB, 아마존 S3
문서 저장 구조	<ul style="list-style-type: none"><li>문서에는 다른 스키마가 있다.</li><li>레코드 간의 관계 설명이 가능</li><li>개념적으로 RDBMS와 비슷</li></ul>	아마존 SimpleDB 아파치 CouchDB MongoDB
컬럼 저장 구조	<ul style="list-style-type: none"><li>연관된 데이터 위주로 읽는 데 유리한 구조</li><li>하나의 레코드를 변경하려면 여러 곳을 수정해야 한다.</li><li>동일 도메인의 열 값이 연속되므로, 압축 효율이 좋다.</li><li>범위 질의에 유리</li></ul>	아파치 Cassandra

### NoSQL의 기술적 특성

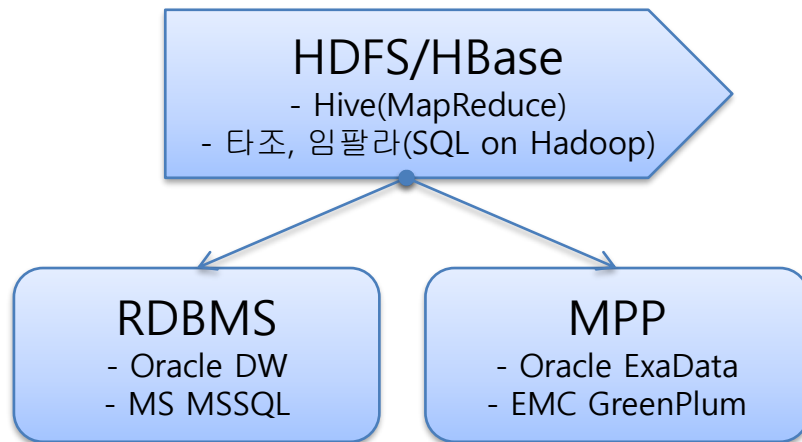
특성	내용
無 스키마	<ul style="list-style-type: none"><li>데이터를 모델링하는 고정된 데이터 스키마 없이 키Key 값을 이용하여 다양한 형태의 데이터 저장 및 접근 가능</li><li>데이터 저장 방식은 크게 열(Column), 값(Value), 문서(Document), 그래프(Graph) 등의 네 가지를 기반으로 구분</li></ul>
탄력성	<ul style="list-style-type: none"><li>시스템 일부에 장애가 발생해도 클라이언트가 시스템에 접근 가능</li><li>응용 시스템의 다운 타임이 없도록 하는 동시에 대용량 데이터의 생성 및 갱신</li><li>질의에 대응할 수 있도록 시스템의 규모와 성능 확정이 용이하며, 입출력의 부하를 분산시키는 데도 용이한 구조</li></ul>
질의(Query) 기능	수십 대에서 수천 대 규모로 구성된 시스템에서도 데이터의 특성에 맞게 효율적으로 데이터를 검색·처리할 수 있는 질의 언어, 관련 처리 기술, API 제공
캐싱(Caching)	<ul style="list-style-type: none"><li>대규모 질의에도 고성능 응답 속도를 제공할 수 있는 메모리 기반 캐싱 기술을 적용하는 것이 중요</li><li>개발 및 운영에도 투명하고 일관되게 적용할 수 있는 구조</li></ul>

# 5. 하이브리드 DW

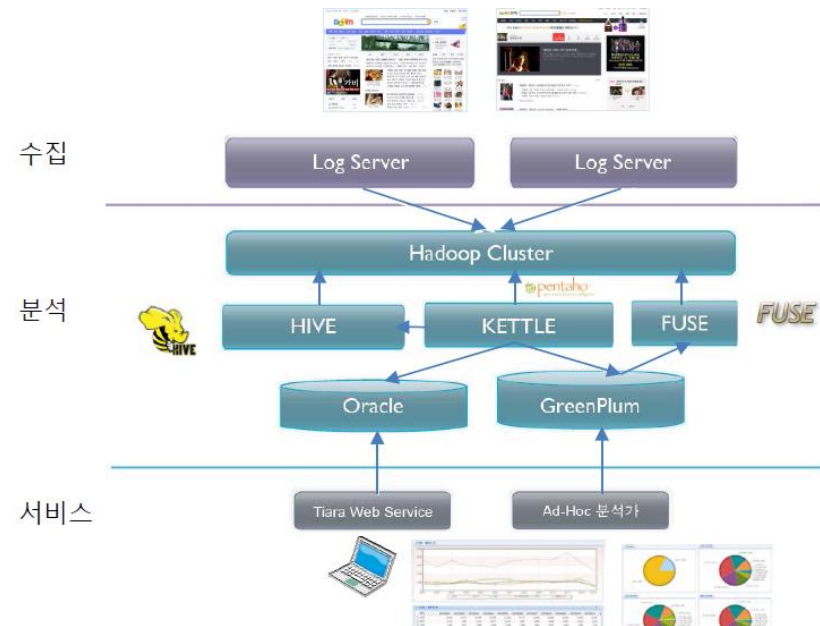


## ● 하이브리드 DataWarehouse

하이브리드 DW 구성 및 SQL 쿼리 도구



- Raw Data  
-> HDFS에 저장
- Summary Data  
-> Hive, RDBMS, MPP에 통합 저장
- Batch Processing  
-> MapReduce/Hive SQL
- Report  
-> RDBMS Query
- Ad-hoc Query  
-> 엑사데이터(제조), 그린플럼(포털)



[다음의 광고 분석 서비스]

## Part III : 수집/저장/처리 기술

# 3.4 빅데이터 처리 기술



# 1. 빅데이터 처리 기술



## ● 분산병렬처리 기술

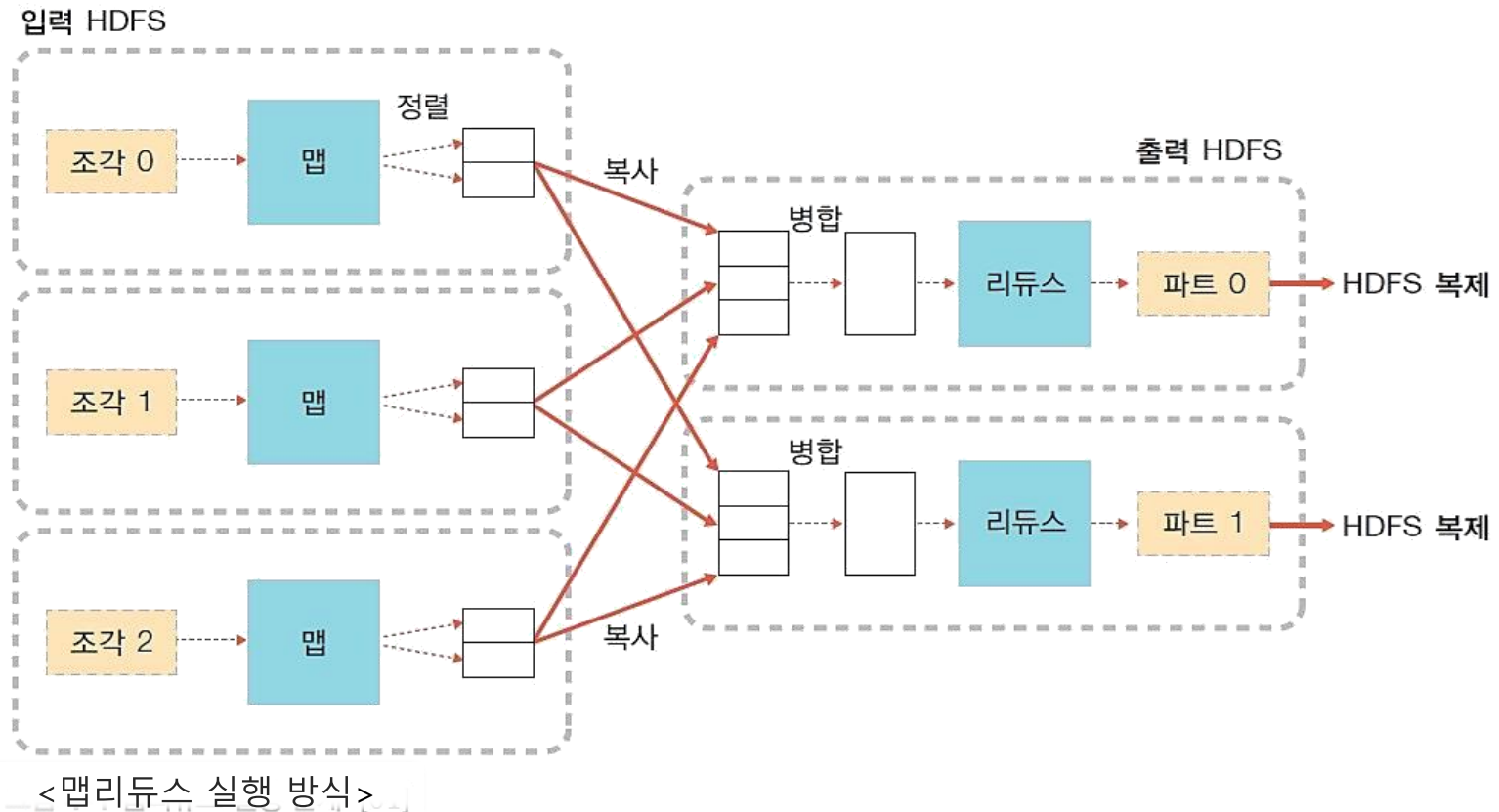
- 하둡 맵리듀스
  - 분산 병렬 데이터 처리 기술의 표준, 일반 범용 서버로 구성된 군집화 시스템을 기반으로 <키,값> 입력 데이터 분할 처리 및 처리 결과 통합 기술, job 스케줄링 기술, 작업 분배 기술, 태스크 재수행 기술이 통합된 분산 컴퓨팅 기술
  - 맵리듀스, Pig, Hive
- SQL on Hadoop
  - 배치처리 중심의 맵리듀스의 한계를 넘기 위해 만들어진 SQL 기반의 자체 쿼리 실행 엔진이다.
  - Hive on Tez, Impala, Presto, Shark(SparkSQL)

## 2. 맵리듀스



### ● 하둡 맵리듀스

- 맵 : 입력(HDFS) -> Emit (Key, Value)
- 셔플 : key로 정렬하여 Reduce로 전송
- 리듀스 : 개별 key로 묶인 Value의 리스트로 연산 수행 -> 결과 저장(HDFS)





# 3. SQL On Hadoop



## ● SQL On Hadoop

- SQL On Hadoop 개요
  - 정의
    - HDFS에 저장된 데이터를 SQL 혹은 SQL과 유사한 형태로 처리를 요청하고 분산 처리하는 시스템
  - 특징
    - 기존 시스템에서 주로 사용한 SQL형식으로 하둡의 데이터를 분산 처리할 수 있다.
  - RDBMS와 SQL on Hadoop의 선택
    - 빠른 처리가 필요한 비정형 쿼리나 OLTP는 RDBMS를 선택
    - 맵리듀스 대신 자체 엔진을 사용한 실시간 쿼리를 최근에 나온 SQL on Hadoop 기술이 지원
- SQL On Hadoop의 장점
  - 개발 시간의 단축
    - 맵리듀스 보다 쉬움, 프로젝트 수행 인력을 구하기 쉽다.
  - 기존 DW의 SQL과 유사하다.
    - 기존 프로그램의 상당수는 SQL로 되어 있음, 처리로직을 이해하는데 수월
    - 마이그레이션할 때 필요한 자원이 크게 줄어든다.
  - 직관성
    - 직관성이 높고 반복 과정에 수월하다. 단계적 쿼리 분석을 통해 유용한 분석 방법과 결과를 도출한다.



# 3. SQL On Hadoop



## ● SQL On Hadoop

- Hive on Tez
  - 호튼웍스의 스팅거 프로젝트 -> Hive를 발전시키고 성능을 강화
  - MapReduce -> 새로운 MapReduce인 Tez를 지원
  - Hive 쿼리 엔진으로 빠른 응답 속도
- 드릴(Drill)
  - 구글의 Dremel(일명 BigQuery)의 오픈소스 버전
  - 높은 확장성과 하둡/NoSQL과의 쿼리 인터페이스 지원이 특징
- SparkSQL(Shark)
  - 스파크(Spark)는 인메모리 기반의 초고속 분산병렬처리 프레임워크.
  - SparkSQL은 Hive에 저장된 데이터에 SQL 질의를 할 수 있는 것이 특징.
  - 초기 버전은 Hive의 소스코드를 일부 변경한 Shark
- Impala
  - Hive 사용자를 고려한 빠른 SQL 쿼리 엔진.
  - HDFS나 HBase에 저장된 데이터를 대상으로 분석, 문법은 HiveQL과 동일
- Presto
  - 페이스북에서 개발
  - 빠른 속도의 검색과 집계 가능

## 4. 실시간 처리 기술



### ● 실시간 처리 기술

CEP	ESP
<ul style="list-style-type: none"><li>- 복합 이벤트의 분석을 통해 패턴에서 중요한 정보를 획득</li><li>- 수직적 확장</li><li>- 다양한 이벤트 분석이 중요</li></ul>	<ul style="list-style-type: none"><li>- 실시간 이벤트 데이터의 빠른 분석을 통해 실시간 정보 제공</li><li>- 수평적 확장</li><li>- 실시간 분석이 중요</li></ul>

<주요 실시간 처리 기술>

구분	세분	내용
CEP	상용 제품	Oracle CEP, IBM Webshere Business Event
	Esper	Java와 .NET을 지원하는 CEP와 이벤트 분석
ESP	Storm	분산 실시간 컴퓨팅 시스템
	S4	실시간 데이터 프로그램 개발을 위한 플랫폼

# 5. Spark Streaming



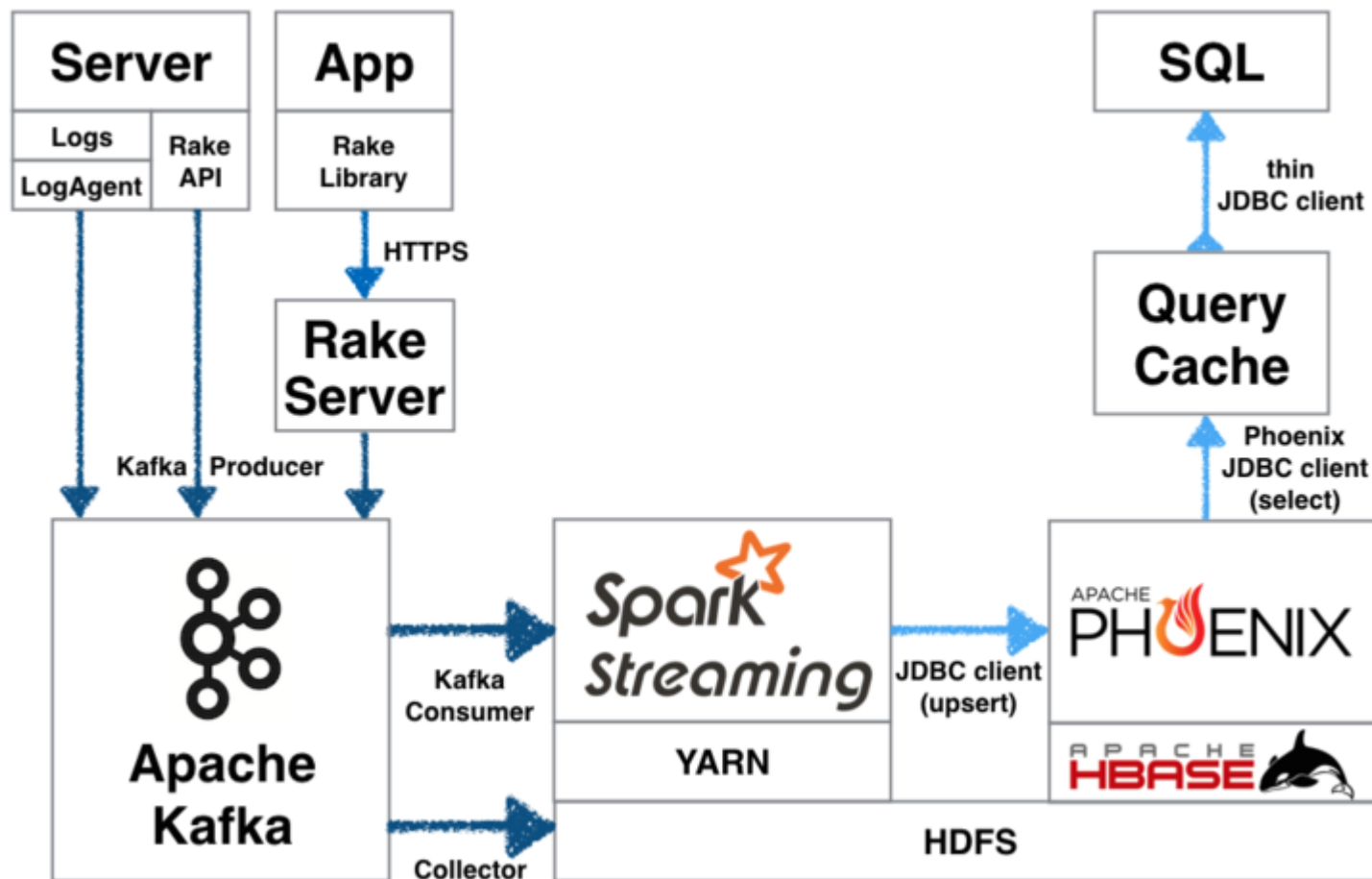
## ● Spark Streaming

- 스파크 스트리밍
  - 개요 : 실시간 스트리밍 데이터 처리
  - DStream : 시간별(특정 시간 간격)로 도착한 데이터들의 연속적인 모임
  - 데이터 소스 : Flume, Kafka, HDFS 등
- 주요 아키텍처
  - 마이크로 배치(Micro-Batch)
    - 작은 배치 단위들 위에서 연속적인 흐름에 따라 처리한다.
  - 트랜스포메이션
    - 무상태 트랜스포메이션
    - 상태유지 트랜스포메이션
    - 윈도우 트랜스포메이션
  - 결과 연산
    - 화면 출력 : Print()
    - 저장 : Save()

# 5. Spark Streaming



- Stream Processing with Spark Streaming



**감사합니다.**