Hadoop and Ecosystem

2016.4

권순범

Kookmin University

목 차

❑ Hadoop

- 대용량 데이터를 분산 처리할 수 있는 Java기반 오픈소스 프레임워크
- Open-source software for reliable, scalable, distributed computing
- Designed to scale up from single server to thousands of machines, each offering local computation and storage

❑ 4 Modules

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™):** A distributed file system that provides high-throughput access to application data.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.

❑ 25 January, 2016: Release 2.7.2 available

❑ History

- 2005년에 Doug Cutting이 구글이 논문으로 발표한 GFS(Google File System)와 맵리듀스(MapReduce)를 구현한 결과물
- 오픈소스 검색 엔진인 너치(Nutch)에 적용하기 위해 시작됐다가 이후 독립적인 프로젝트로 만들어졌고, 2008년에는 아파치 최상위 프로젝트로 승격되었음
- 더그 커팅이 자신의 아들이 노란 코끼리 장난감 인형을 하둡이라고 부르는 것을 듣고 하둡이라는 이름을 지었음. (노란 코끼리 로고)
- 프로젝트 네이밍 – 이후 하둡 관련 서브 프로젝트도 동물과 관련된 이름을 주로 사용

“ 페이스북은 데이터 중 일부를 하둡에 저장한다. 그 양은 약 30PB로, 미국 국회도서관에 저장돼 있는 정보량의 3배에 이른다. 페이스북은 하둡을 이용해 대용량 사진 데이터를 작은 데이터로 쪼개 처리한다. 약 2천여개의 서버가 매 순간 데이터를 처리한다. 사용자가 페이스북에 사진을 쉽게 올리고 내려받으며, 다른 사람의 페이스북 사진을 클릭과 동시에 볼 수 있는 이유다.”

하둡의 특징

❑ 오픈소스

- 하둡은 오픈소스 프로젝트이기에 소프트웨어 라이선스 비용에 대한 부담이 없음
- 많은 벤더가 기능추가, 코드개선을 통해서 독자적 배포판을 개발, 공급
- 주요 배포판: 클라우데라, MapR, 호튼웍스

❑ 분산처리

- 기존 RDBMS는 서버에서 데이터를 처리하는 방식
- 하둡은 여러 대의 서버에 데이터를 저장하고, 데이터가 저장된 각 서버에서 동시에 데이터를 처리하는 방식임

❑ 데이터복구 처리

- 하둡은 데이터의 복제본을 저장하기 때문에 데이터의 유실이나 장애가 발생했을 때도 데이터의 복구가 가능함

하둡의 특징

❑ 비용

- 값비싼 유닉스 장비 대신 x86 CPU에 리눅스 서버에 설치하고 운영할 수 있음.
- 데이터 저장 용량이 부족할 경우, 필요한 만큼 리눅스 서버만 추가하면 됨

❑ 장점

- 비용 대비 빠른 데이터 처리
- 장애를 대비한 특성

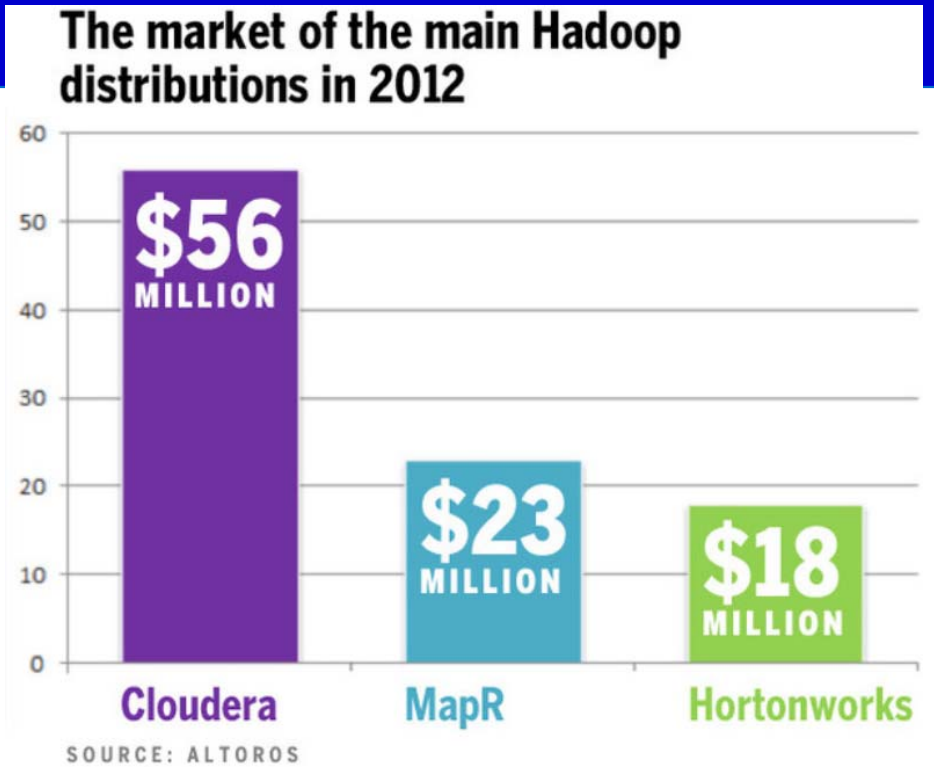
❑ 활용기업의 확대

- 초기에 야후에서 주도적으로 사용
- 아마존, 이베이, 페이스북 등 글로벌 서비스 업체
- 국내 NHN, DAUM과 같은 포털 기업과 KT, SKT(2008년 도입/사용) 같은 통신업체

Hadoop Distributions

- 하둡의 알려진 버그만 수천개?

- 주요 벤더 배포판



– **신뢰성**. 벤더들은 버그가 발견됐을 때 더 빨리 대응을 한다. 즉시 픽스와 패치를 배포해, 솔루션을 더 안정적으로 만든다.

– **지원**. 기업 사명 달성에 아주 중요한 플랫폼 도입과 엔터프라이즈급 작업 처리가 가능하도록 기술 지원 서비스를 제공하는 회사들이 많다.

– **완성도**. 특정 작업을 처리하기 위해, 다른 툴로 기능을 보강한 하둡 배포판이 아주 많다. 여기에 더해, 벤더들은 하둡 표준 배포판을 개선하는데 공헌하고 있다. 오픈소스 저장소에 업데이트한 코드를 제공하면서 하둡 공동체의 성장을 촉진하고 있는 것이다.

Hadoop Distributions

❑ Distributions and Commercial Support

- Hadoop Wiki (http://wiki.apache.org/hadoop/FrontPage)
- Distributions List
  http://wiki.apache.org/hadoop/Distributions%20and%20Commercial%20Support?action=show&redirect=Distribution

❑ Hadoop Conference

- Hadoop Summit (봄)
- Hadoop World (가을)

하둡의 특징

❑ 벤치마크

- https://amplab.cs.berkeley.edu/benchmark/

❑ Quantitative and Qualitative comparisons of five systems:

- **Redshift :** a hosted MPP database offered by Amazon.com based on the ParAccel data warehouse.
- **Hive:** a Hadoop-based data warehousing system. (v0.12)
- **Shark :** a Hive-compatible SQL engine which runs on top of the **Spark** (v0.8.1)
- **Impala :** a Hive-compatible* SQL engine with its own MPP-like execution engine. (v1.2.3)
- **Stinger/Tez :** Tez is a next generation Hadoop execution engine currently in development (v0.2.0)

1. Scan Query

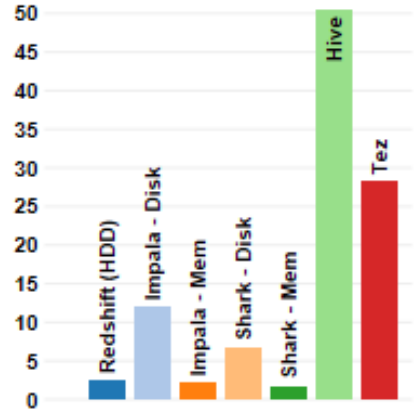
```
SELECT pageURL, pageRank FROM rankings WHERE pageRank > X
```

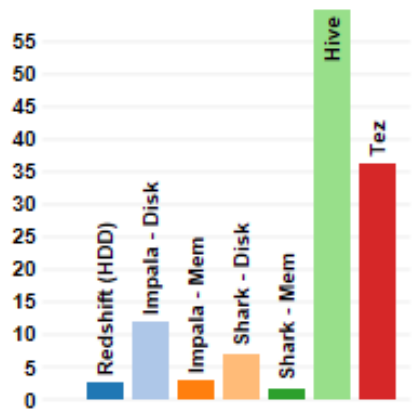
Query 1A
32,888 results

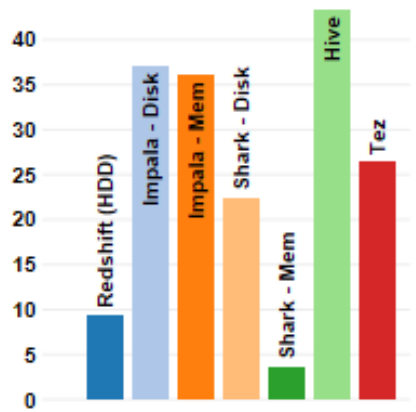
Query 1B
3,331,851 results

Query 1C
89,974,976 results

Response Time







하둡의 연계확장

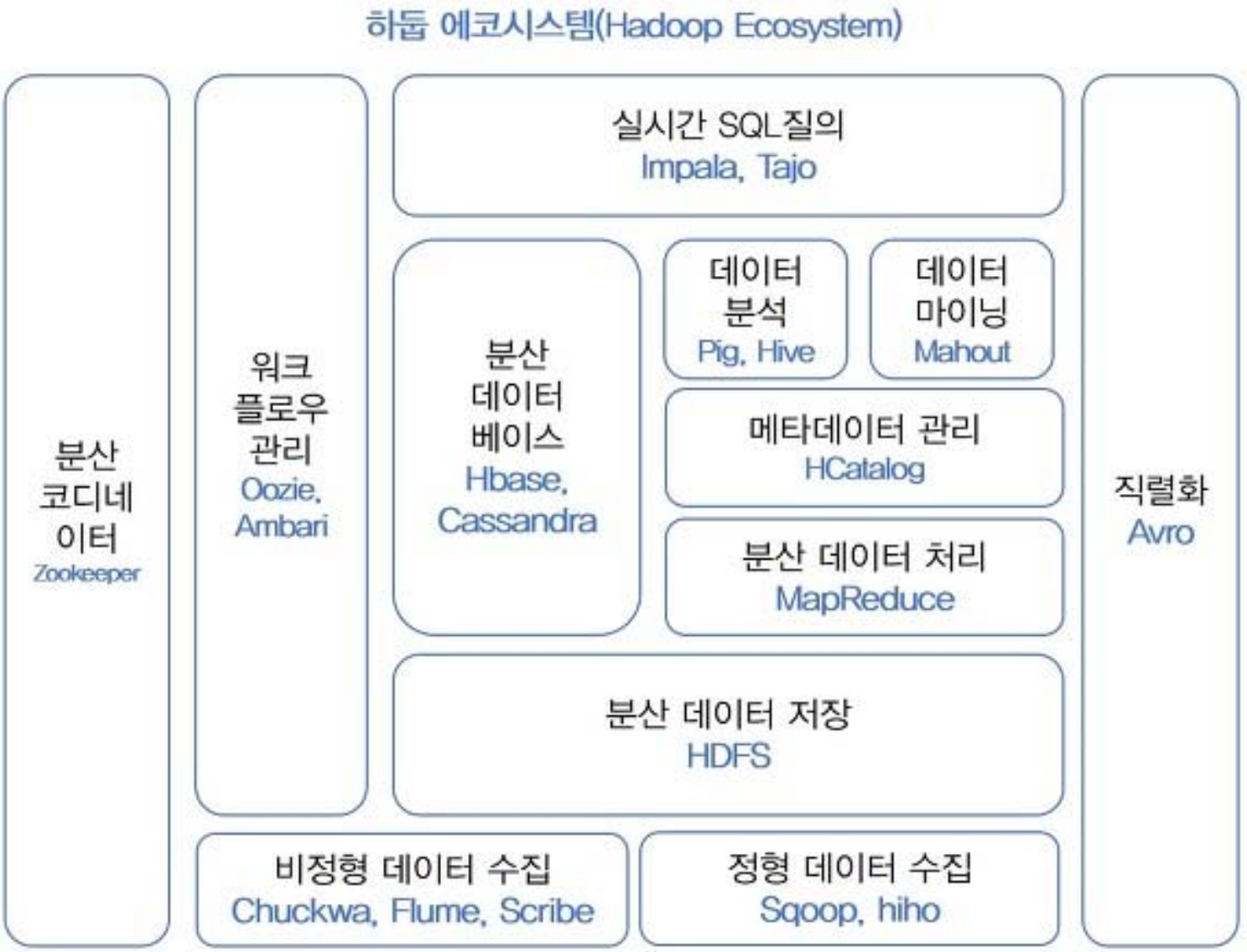
❑ 하둡은 빅데이터 플랫폼의 핵심 기술이자 사실상 표준

- 빅데이터 플랫폼의 진화
  - 실시간 데이터 처리
  - 다양한 방법의 분산병렬처리
  - 관계형 데이터 모델 지원 등의 방향

빅데이터 플랫폼의 한계와 진화 방향

빅데이터 특성	한계 및 제약	진화 방향
용량 (Volume)	· 데이터센터 규모의 관리 한계	· 범 지구적 규모로 확장 가능 (지역 거점 별 데이터센터 확장)
다양성 (Variety)	· 특정 분산병렬 처리 방법 제공 · 관계형 데이터 모델 미지원	· 다양한 분산병렬 처리 방법 제공 · 관계형 데이터베이스 및 트랜잭션 제공
속도 (Velocity)	· 일괄처리(실시간 처리 미지원)	· 실시간 데이터 조회와 처리

Hadoop Ecosystem



하둡 코어 프로젝트: **HDFS, MapReduce**

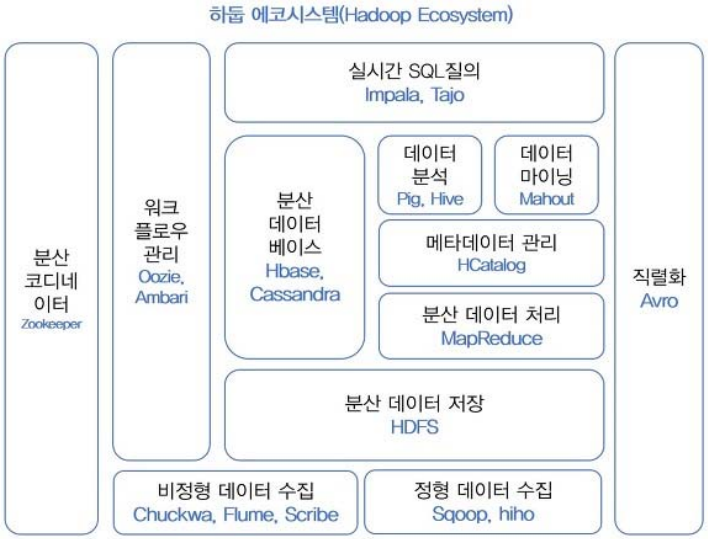
Hadoop Ecosystem with Google

표-2. 구글 논문과 하둡, 에코시스템 비교

구분	구글 논문	하둡과 에코시스템	설명
파일 시스템	2003 년 GFS	2006 년 HDFS	분산 파일 시스템
	2010 년 Colossus	- -	GFS 단점 보완
데이터 처리	2004 년 MapReduce	2006 년 MapReduce	분산병렬처리
	2005 년 Sawzall	2008 년 Pig, Hive	빅데이터 쿼리(일괄처리)
	2009 년 Pregel	개발 중 Giraph	대규모 그래프 데이터 처리
	2010 년 Dremel	개발 중 Drill, Impala	실시간 빅데이터 쿼리
데이터베이스	2006 년 BigTable	2008 년 HBase	Schemaless NoSQL DB
	2012 년 Spanner	- -	빅데이터 트랜잭션 처리
서버 관리	2006 년 Chubby	2008 년 Zookeeper	서버 락(Lock) 서비스

하둡 에코시스템 (2)

❑ 분산 데이터를 저장하는 HDFS와 분산 데이터를 처리하는 맵리듀스가 하둡 코어 프로젝트에 해당하며 나머지는 모두 하둡의 서브 프로젝트



❑ Zookeeper(분산 코디네이터)

- 분산 환경에서 서버 간의 상호 조정이 필요한 다양한 서비스 제공

❑ Oozie(워크플로우 관리)

- 하둡 작업을 관리하는 워크플로우 및 코디네이터 시스템

❑ HBase(분산 데이터베이스)

- HDFS 기반의 컬럼 데이터베이스로 실시간 랜덤 조회 및 업데이트가 가능함
- 구글의 BigTable 논문을 기반으로 개발됨.
- NHN이 모바일 메신저 라인에 HBase를 적용한 시스템 아키텍처를 발표했음

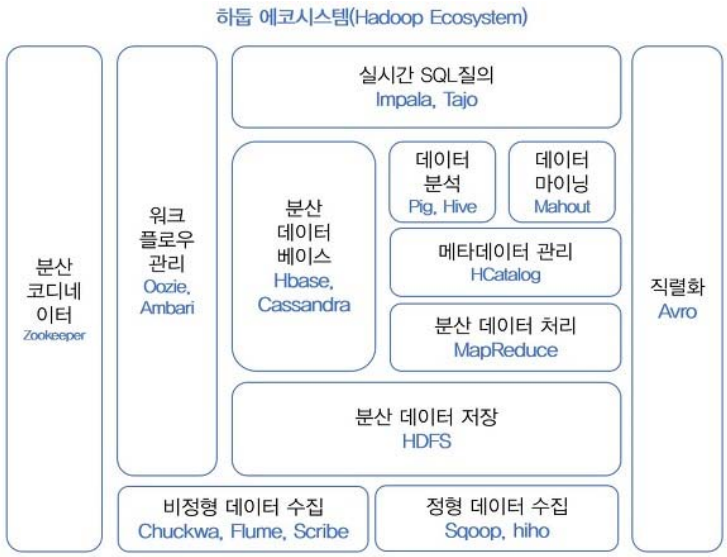
하둡 에코시스템 (3)

❑ Pig

- 데이터 분석을 위한 하이레벨 스크립트 언어(야후에서 개발)
- 복잡한 맵리듀스 프로그래밍을 대체할 Pig Latin 이라는 자체 언어를 제공
- 맵리듀스 API를 크게 단순화하여 SQL 과 유사한 형태로 설계

❑ Hive(데이터 분석)

- 페이스북에서 개발된 하둡 기반의 DW용 솔루션임.
- SQL과 매우 유사한 HiveQL 이라는 쿼리를 제공함.
- HiveQL은 내부적으로 맵리듀스 잡으로 변환되어 실행됨



❑ Mahout(데이터 마이닝)

- 데이터 마이닝 알고리즘을 구현한 오픈소스임.
- Classification, clustering, 추천, 협업 필터링, 회귀 분석 등을 지원함

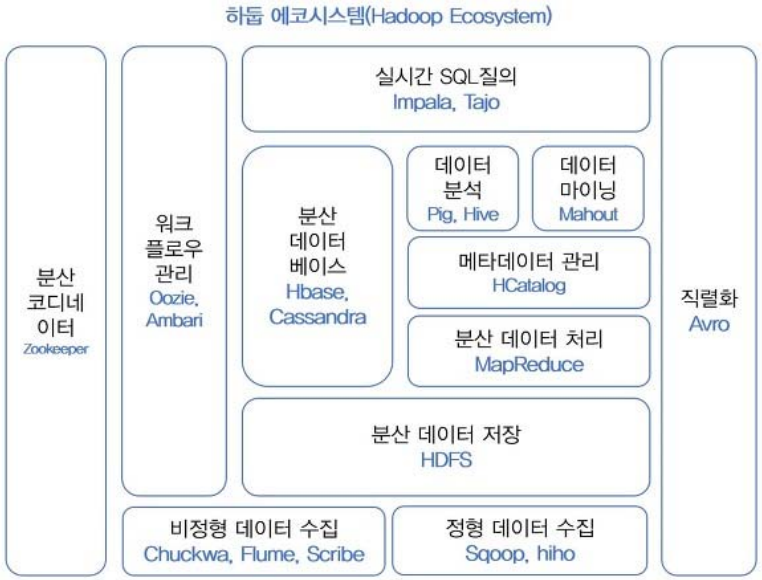
하둡 에코시스템 (4)

❑ Hcatalog (메타데이터 관리)

- 하둡으로 생성한 데이터를 위한 테이블 및 스토리지 관리 서비스

❑ Avro(직렬화)

- RPC(Remote Procedure Call)와 데이터 직렬화를 지원하는 프레임워크.
- JSON을 이용해 데이터 형식과 프로토콜을 정의하며 작고 빠른 바이너리 포맷으로 데이터를 직렬화 함



하둡 에코시스템 (5)

❑ Chukwa(비정형 데이터 수집)

- 분산 환경에서 생성되는 데이터를 HDFS에 안정적으로 저장하는 플랫폼
- 분산된 각 서버에서 에이전트를 실행하고 콜렉터가 에이전트로부터 데이터를 받아 HDFS에 저장함

❑ Flume(비정형 데이터 수집)

- Chukwa 처럼 분산된 서버에 에이전트가 설치되고, 에이전트로부터 데이터를 전달받는 콜렉터로 구성
- 차이점은 전체 데이터의 흐름을 관리하는 마스터 서버가 있어서 데이터를 어디서 수집하고 어떤 방식으로 전달할지를 동적으로 변경할 수 있음. (클라우데라에서 개발)

❑ Scribe(비정형 데이터 수집)

- 페이스북에서 개발한 데이터 수집 플랫폼
- Chuckwa 와는 다르게 중앙 집중 서버로 전송하는 방식을 사용함

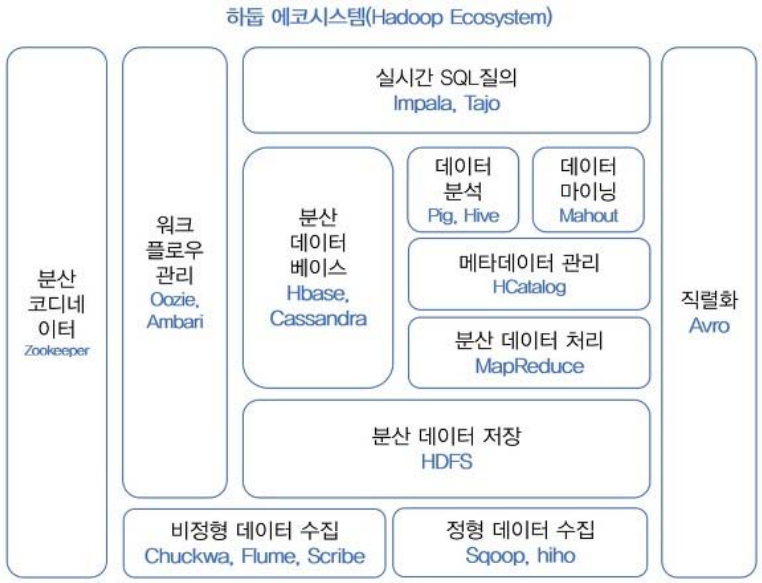
하둡 에코시스템 (6)

❑ Scoop(대용량 데이터 전송 솔루션)

- HDFS, RDBMS, DW, NoSQL 등 다양한 저장소에 대용량 데이터를 신속하게 전송할 수 있는 방법을 제공함

❑ Hiho(대용량 데이터 전송 솔루션)

- Sqoop과 같은 대용량 데이터 전송 솔루션이며 GitHub에 공개돼 있음.
- 오라클과 MySQL의 데이터 전송만 지원함



하둡 에코시스템 (7)

❑ Impala (실시간 데이터조회)

- 클라우데라에서 개발한 하둡 기반의 실시간 SQL 질의 시스템
- 맵리듀스를 사용하지 않고 자체 개발 엔진을 사용해 빠른 성능을 보여줌.
- 데이터 조회를 위한 인터페이스로 HiveQL을 사용하며 Hbase와도 연동할 수 있음

❑ Tajo (실시간 데이터조회)

- 고려대학교 박사 과정 학생들이 주도해서 개발한 하둡 기반의 SW 시스템
- 2013년 아파치 재단의 인큐베이션 프로젝트로 선정됨
- 데이터 저장소는 HDFS를 사용하되, SQL 언어를 사용하여 실시간으로 데이터를 조회할 수 있음

NoSQL (Not only SQL)

❑ NoSQL

- 관계형 데이터 모델과 SQL문을 사용하지 않는 데이터베이스시스템 혹은 저장소
- 기존 RDBMS가 비정형 데이터 저장과 분산 환경에 적합하지 않기에 고안됨

❑ 특징

- 스키마가 없다. 즉, 데이터 관계와 정해진 규격 (table – column 의 정의)이 없다.
  - 관계정의가 없으니 Join이 불가능 하다.
- 분산처리 (수평적 확장) 기능을 제공한다.
  - 샤딩(Sharding)이라는 기능을 이용해 데이터를 분할해서 다른 서버에 나눠 저장함

❑ 솔루션

- MongoDB, HBase, CouchDB, Casandra, Redis 등 다양한 NoSQL 솔루션
  - 국내에서는 MongoDB가 가장 많이 사용됨

NoSQL (Not only SQL)

❑ 종류

- **Key value DB**

Key와 Value의 쌍으로 데이터가 저장되는 유형으로 Amazon의 Dynamo Paper에서 유래되었습니다. Riak, Vodemort, Tokyo 등의 제품이 많이 알려져 있습니다.

- **Wide Columnar Store**

Big Table DB라고도 하며, Google의 BigTable Paper에서 유래되었습니다. Column Family 데이터 모델을 사용하고 있고, HBase, Cassandra, Hypertable이 이에 해당합니다.

- **Document DB**

Lotus Notes에서 유래되었으며, JSON, XML과 같은 Collection 데이터 모델 구조를 채택하고 있습니다. Mongo DB, Cough DB가 이 종류에 해당합니다.

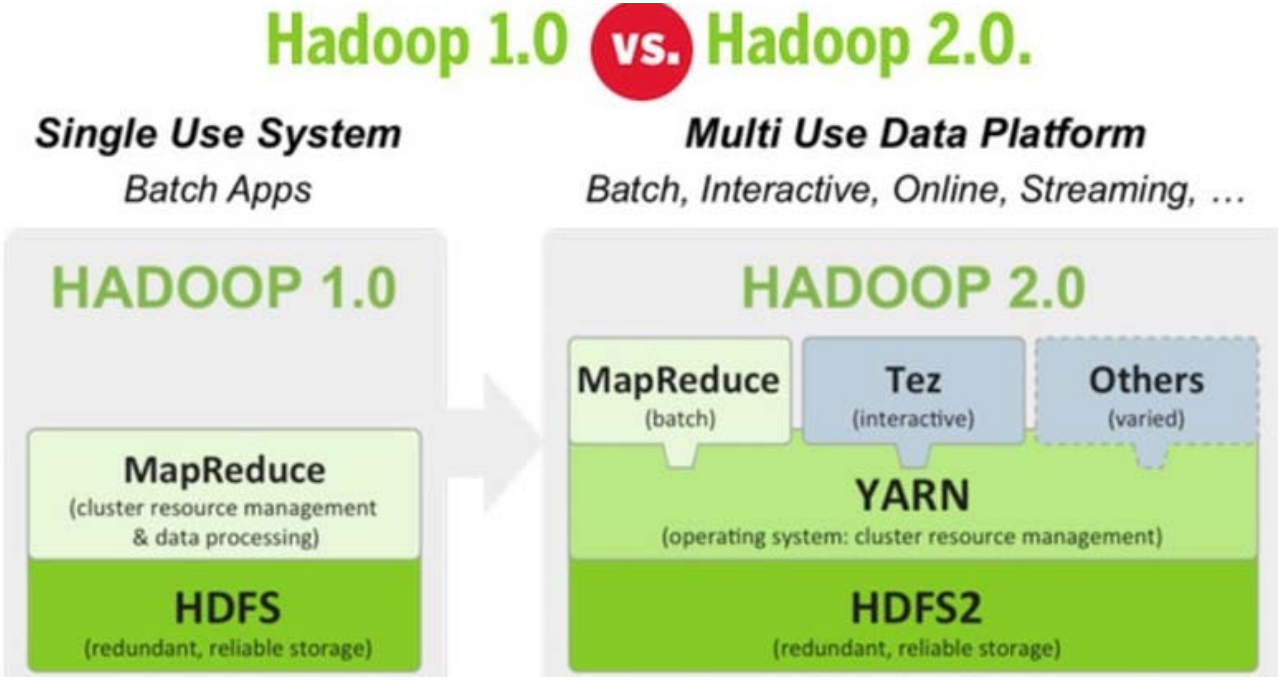
- **Graph DB**

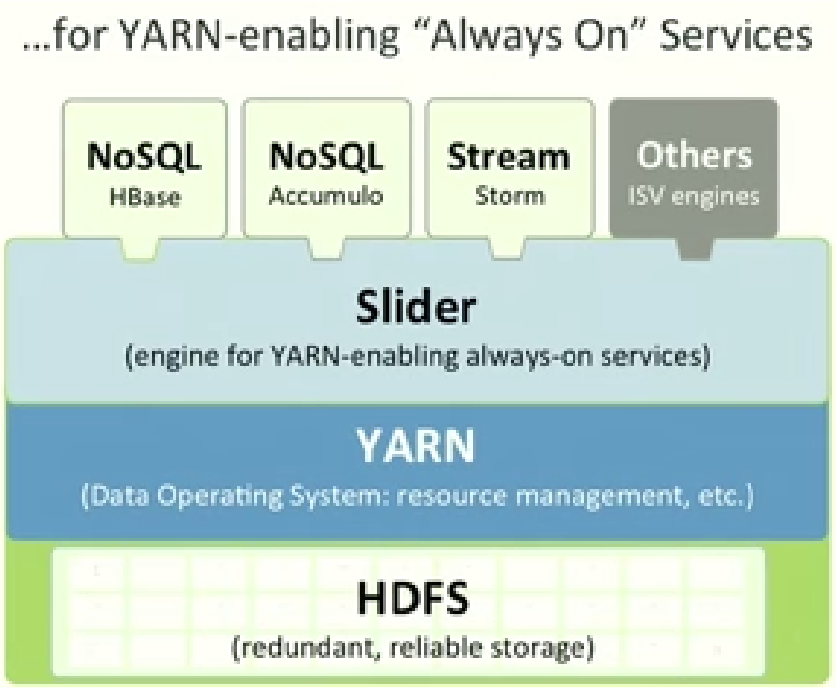
Euler & Graph Theory에서 유래한 DB입니다. Nodes, Relationship, Key-Value 데이터 모델을 채용하고 있습니다. Neo4J, OreientDB 등의 제품이 있습니다.

❑ 장단점

- Flexible한 스키마 처리
- Data governance의 어려움.

Hadoop Ecosystem

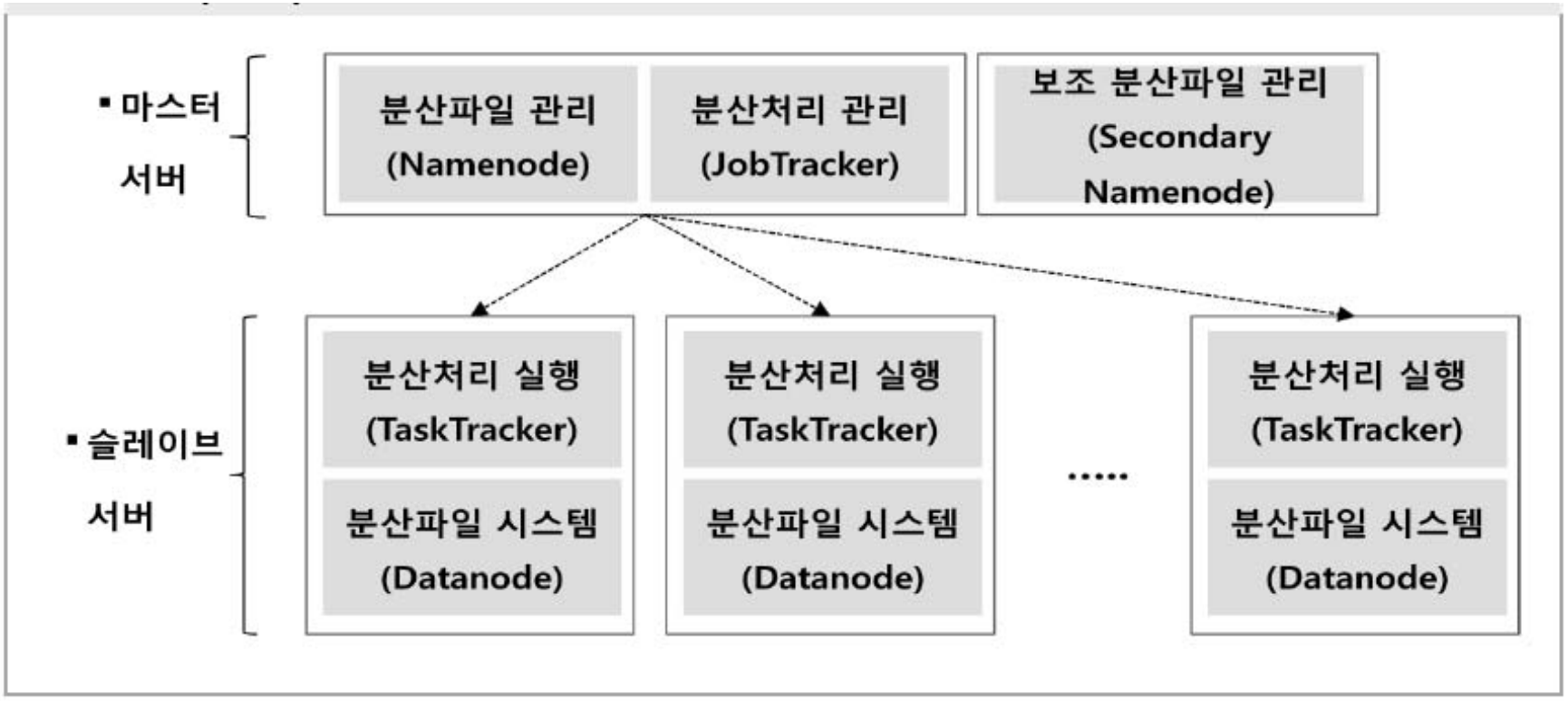




Hadoop 1.0 한계점

❑ 모든 작업은 Job Tracker라는 단일 데몬을 통한 배치 프로세스로 실행

- 확장성 문제
- 프로세싱 병목과 속도 문제



Hadoop 1.0 한계점

❑ **실시간 데이터 처리 한계**

· 하둡 MapReduce 는 일괄처리(Batch) 방식이기 때문에 실시간 데이터 처리, 조회가 안됨

❑ **다양한 데이터 처리 한계**

· 대규모 계산, 데이터간 통신 및 무결성 보장이 필요한 복잡한 연산 및 처리가 어려움

❑ **다수의 작은 파일 관리 어려움**

· 64 메가바이트(MB) [2] 이하의 작은 파일 저장 시 컴퓨팅 자원(메모리) 낭비 초래

❑ **비효율적인 데이터 백업 관리**

· 3 개의 복제본 파일 관리 방식으로 디스크 공간 낭비와 파일 저장에 낮은 성능

· 스냅샷(Snapshot) 방식, 재해복구(DR, Disaster Recovery) 시스템 등의 고급 데이터 백업 미지원

Hadoop 1.0 한계점

❑ **단일고장점(SPoF, Single Point of Failure) 존재**

· 하둡에 저장된 데이터의 메타 정보를 관리하는 마스터 서버의 이중화 미지원으로 마스터 서버 장애 발생시 하둡 전체 시스템 중단

- 마스터 서버 장애에 대한 수동 복구는 가능

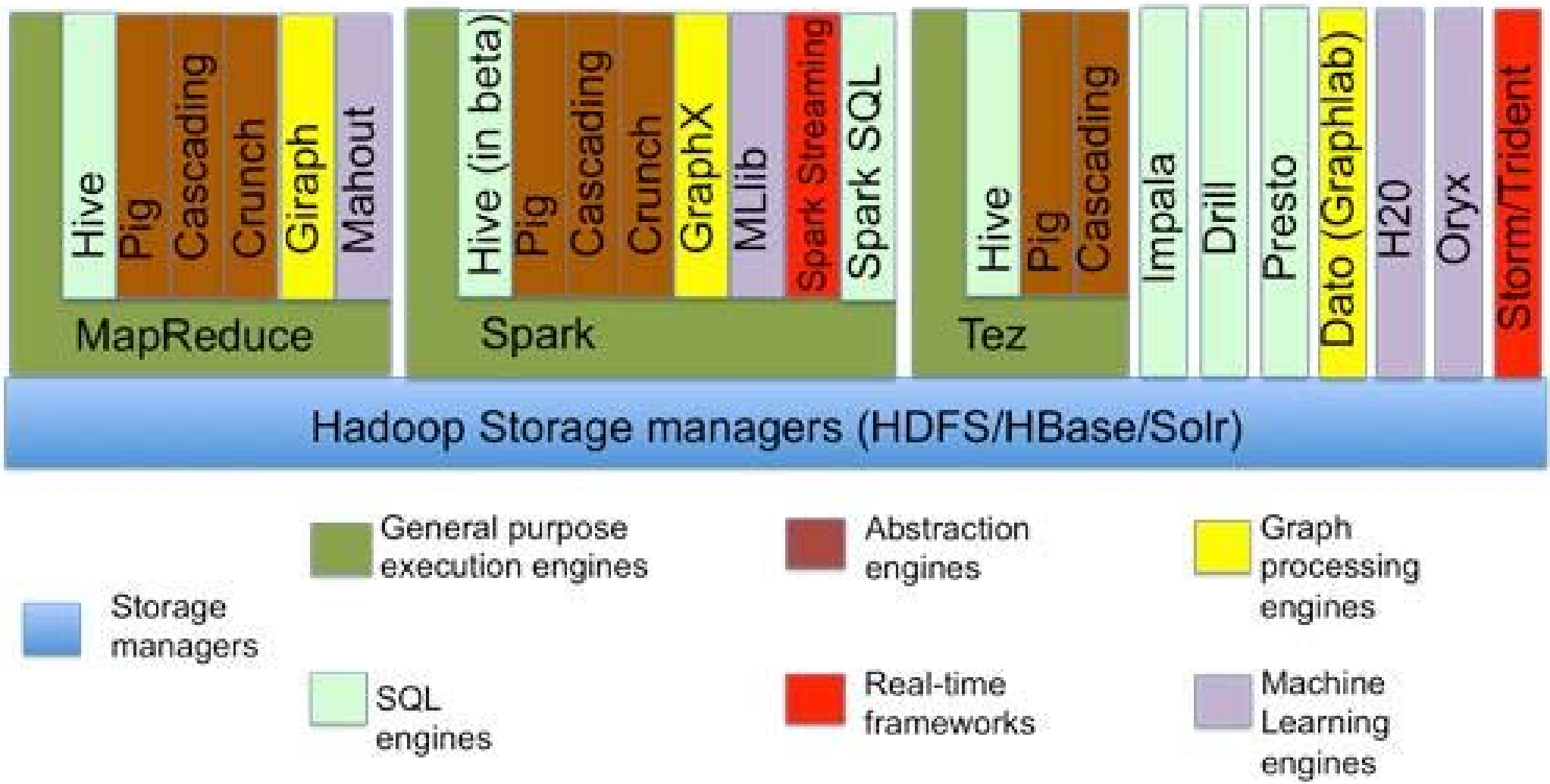
❑ **높은 기술적 숙련도가 요구되고 유지보수가 어려움**

· MapReduce 사용에 컴퓨터 프로그래밍 스킬이 요구됨

- MapReduce 대안으로 하둡의 에코시스템 중 하나인 하이브(Hive)를 활용하여 제한적으로 유사 SQL 방식의 데이터 처리 가능

· 오픈소스 활용 시 기술지원, 유지보수를 스스로 해결해야 하며, 하둡 전문인력 확보가 필요함

Hadoop Ecosystem



Hadoop Ecosystem

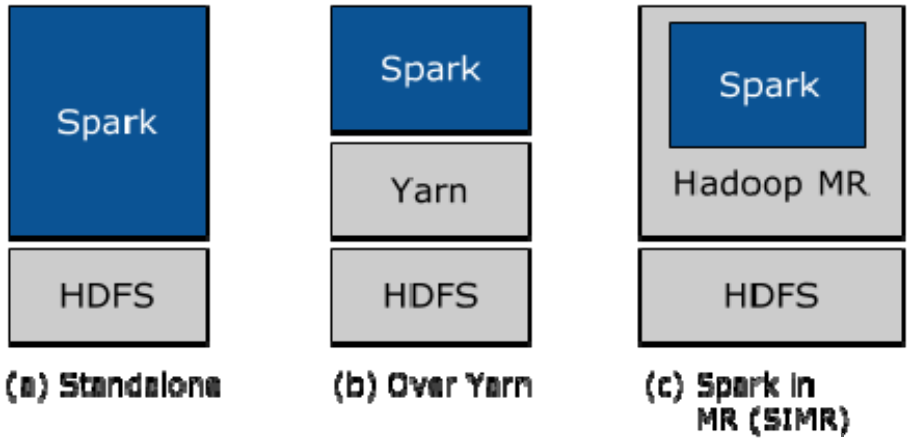
❑ Tez™

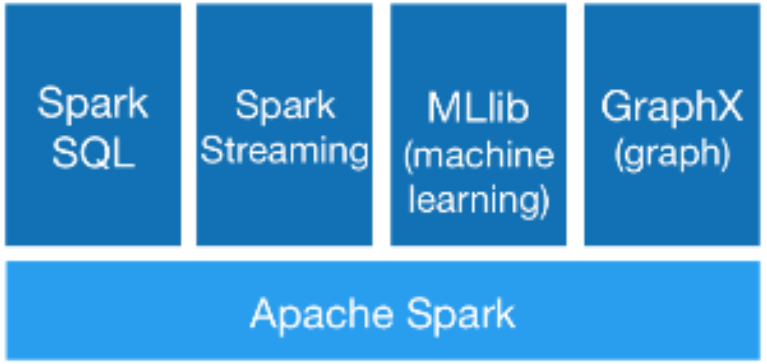
- A generalized data-flow programming framework, built on Hadoop YARN
- Provides a powerful and flexible engine to execute tasks to process data for both batch and interactive use-cases
- Adopted by Hive™, Pig™ and other frameworks in the Hadoop ecosystem and also by other commercial software (e.g. ETL tools), to replace Hadoop™ MapReduce as the underlying execution engine

❖ 맵리듀스는 **Map** 처리와 **Reduce** 처리 프로세스 마다 디스크 쓰기가 발생하지만 **Tez**와 **Spark**에서는 **In-memory** 처리가 가능해 스트림 처리, 반복계산이 많은 기계학습 처리 및 그래프 처리 등의 속도를 높일 수 있는 것이 특징이다.

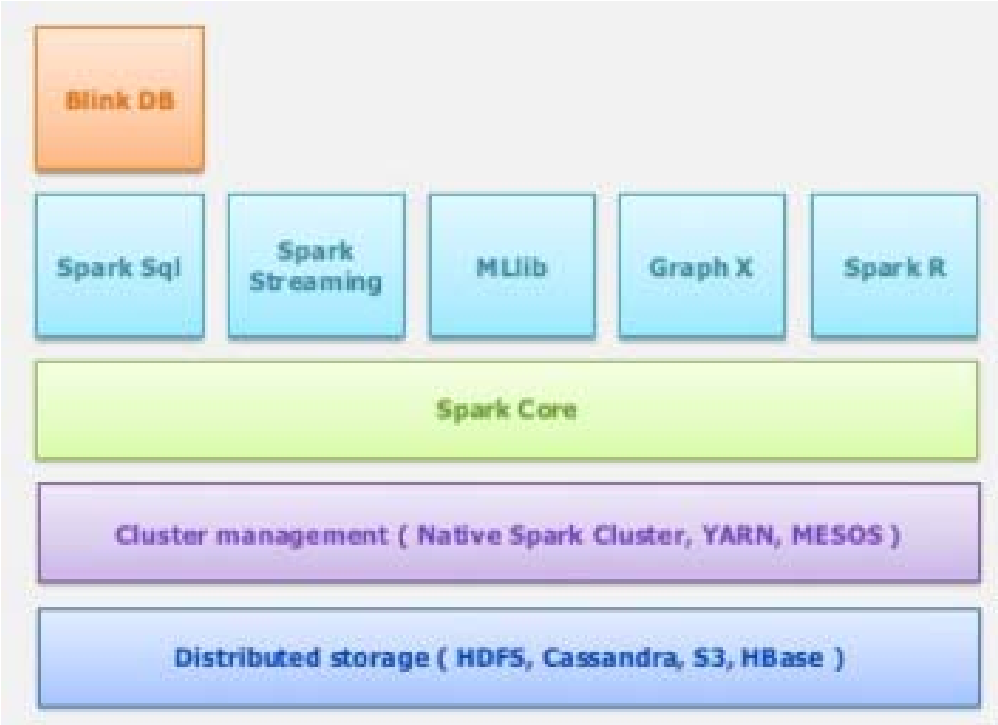
❑ Spark

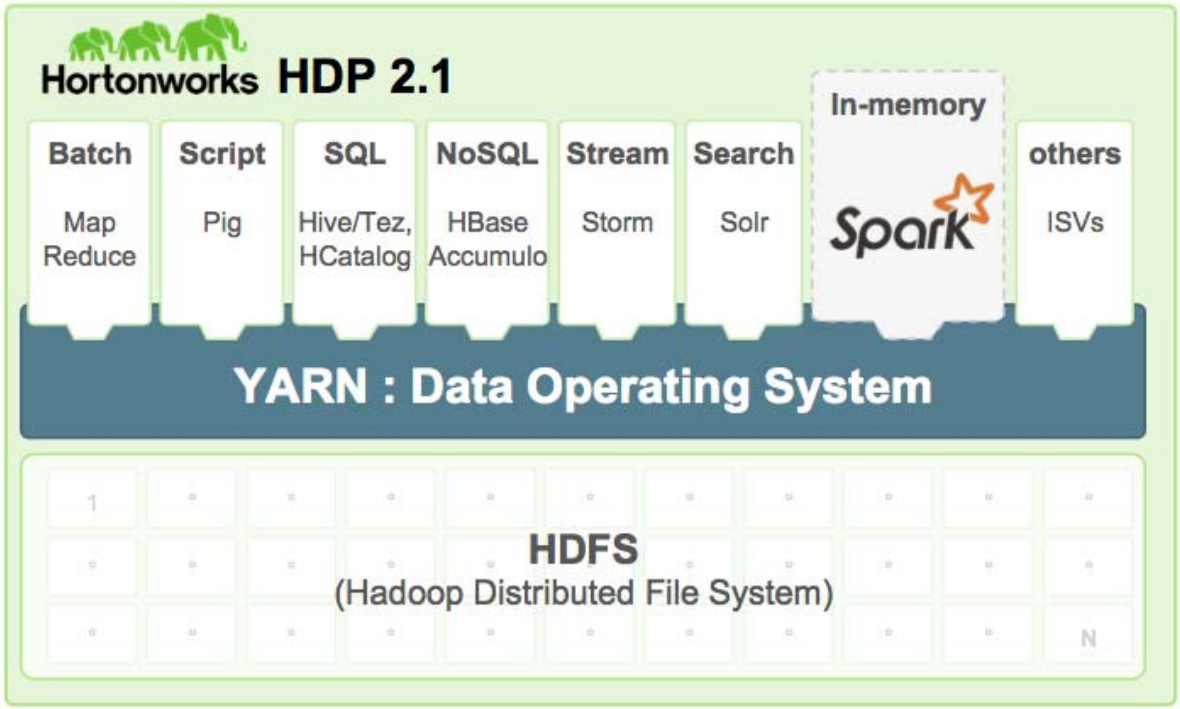
- 빅데이터를 위한 실시간 분산형 컴퓨팅 프로젝트 (2016.1 version 1.6)
- 인메모리 처리를 기본으로 함
- 스파크 실행엔진은 하둡과 같이 일괄 처리를 담당하며 동시에 다른 프레임워크의 기반이 됨
- 스파크는 하둡에서 맵리듀스의 기능을 대체할 수 있음
  - 스파크는 기존 하둡 클러스터에서 실행시킬 수 있으며, 자원 스케줄링에는 얀(YARN)을 이용함

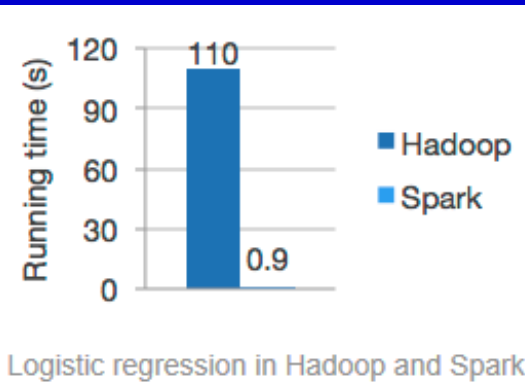




Hadoop Ecosystem





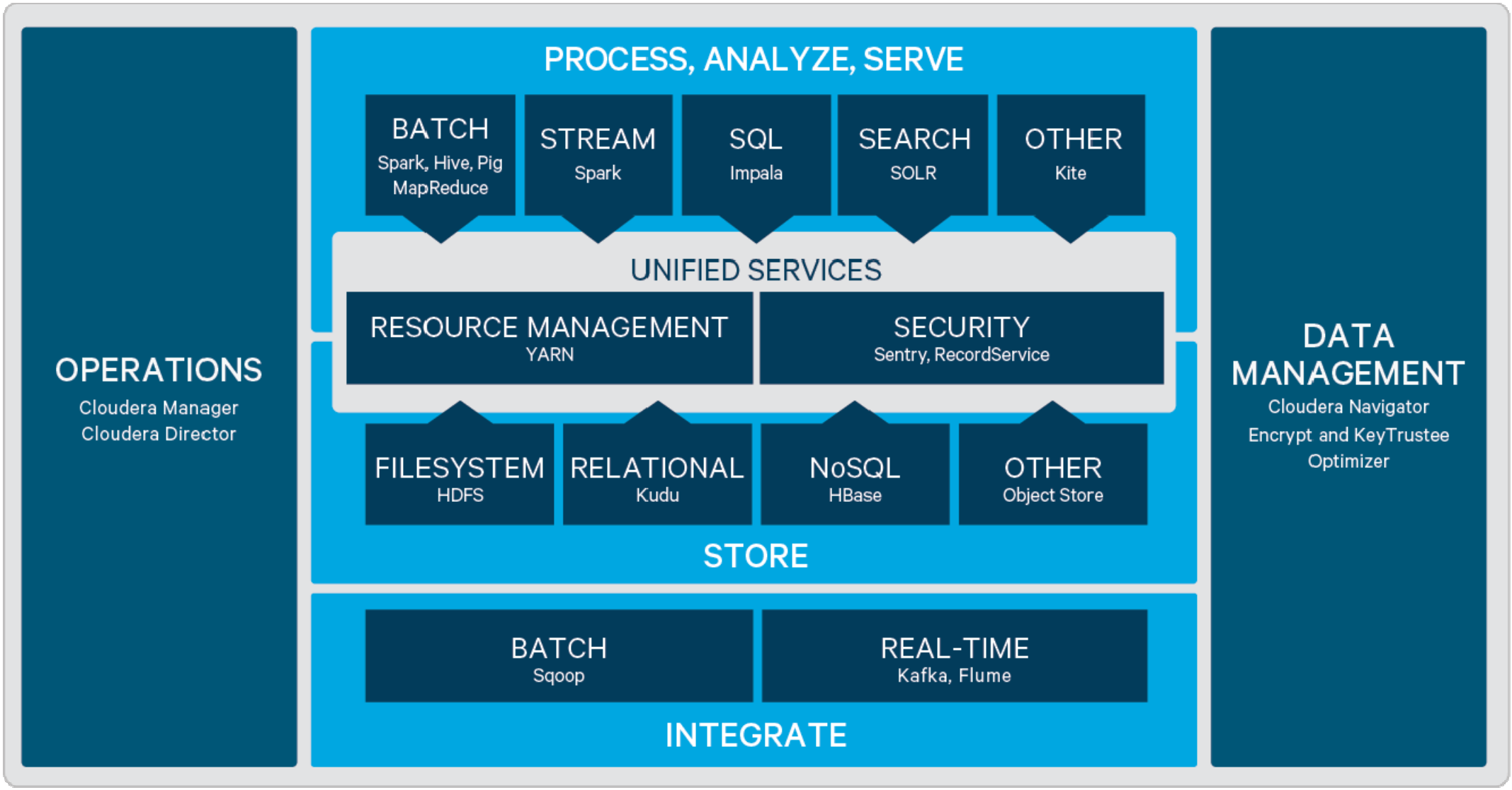


Logistic regression in Hadoop and Spark

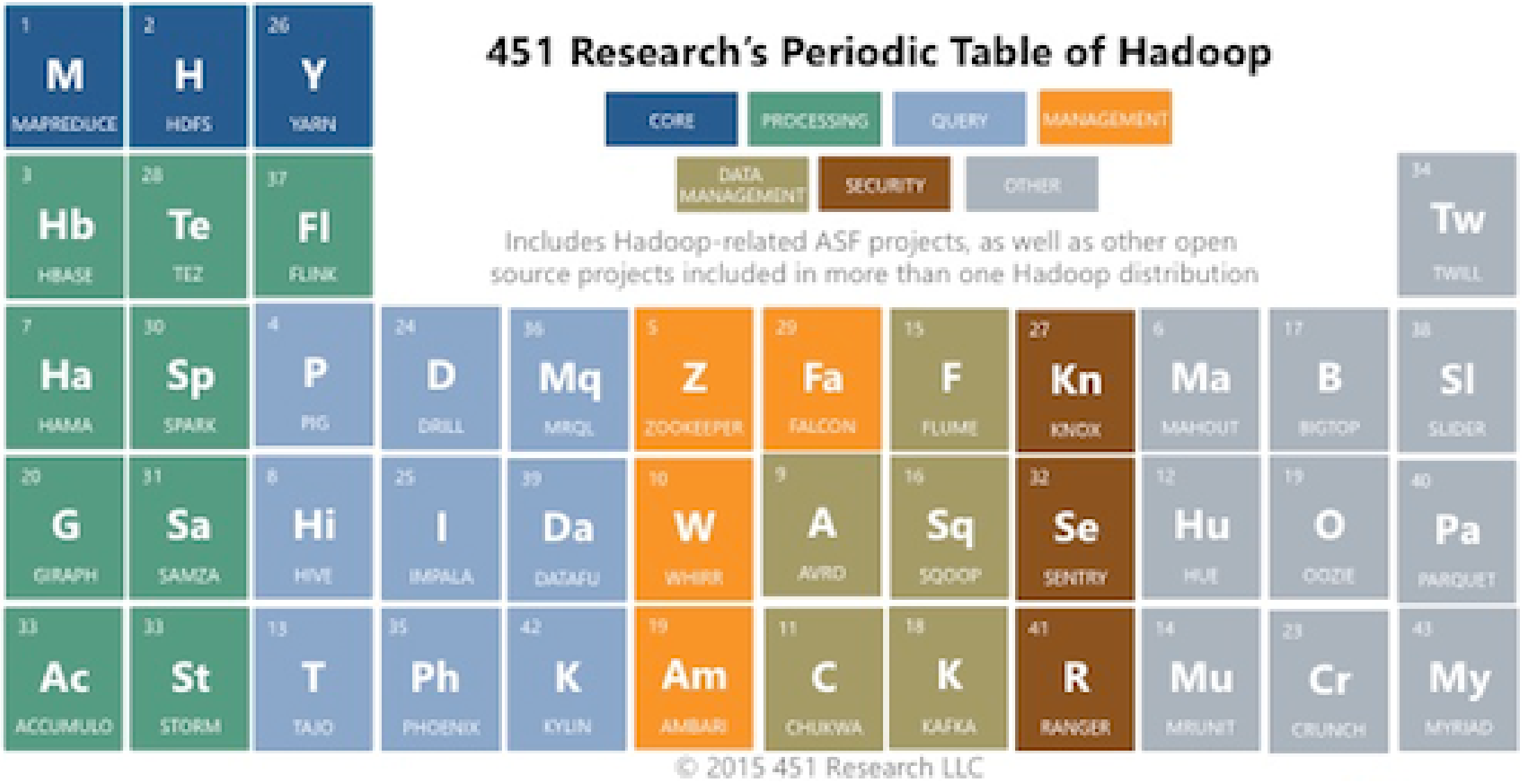
❑ Spark의 장점

- **Runs Everywhere**
  - Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3
- **Generality**
  - Combine SQL, streaming, and complex analytics.
  - Spark powers a stack of libraries: SQL and DataFrames, MLlib, GraphX, and Spark Streaming.
  - You can combine these libraries seamlessly in the same application.
- **Ease of Use**
  - Write applications quickly in Java, Scala, Python, R.
  - Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it interactively from the Scala, Python and R shells.
- **Speed**
  - Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

Hadoop Ecosystem - Cloudera



Combination of More Than 40 Projects



- ❖ 모든 문제를 해결하는 하나의 처리 프레 임워크는 없다는 것이다.
- ❖ 복잡함은 괴로움을 안겨주는 동시에 광활한 미래를 이끌고 있다.

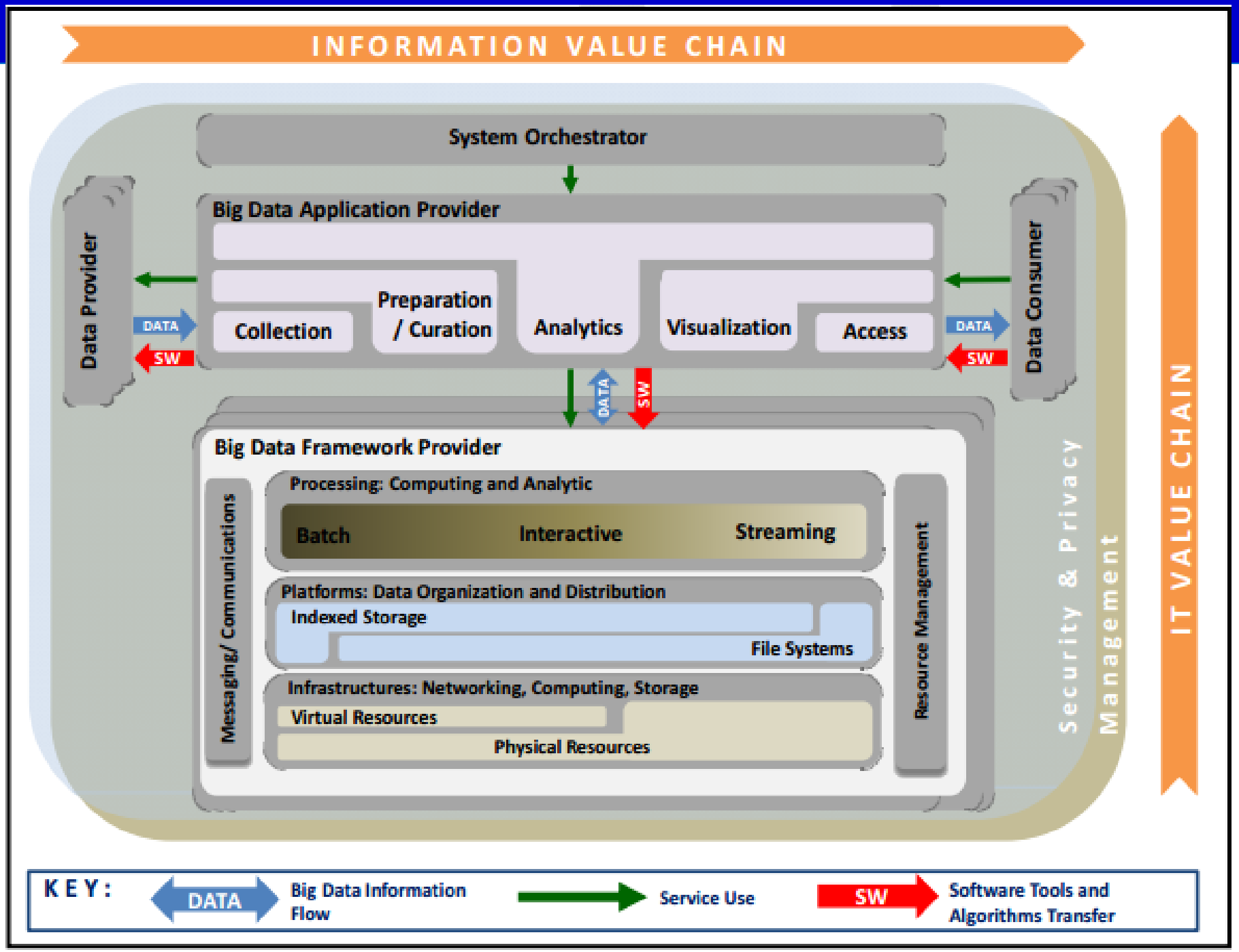


Figure 2: NIST Big Data Reference Architecture (NBDRA)

NIST Big Data Reference Architecture

- **System Orchestrator**: Defines and integrates the required data application activities into an operational vertical system
- **Data Provider**: Introduces new data or information feeds into the Big Data system
- **Big Data Application Provider**: Executes a data lifecycle to meet security and privacy requirements as well as System Orchestrator-defined requirements
- **Big Data Framework Provider**: Establishes a computing framework in which to execute certain transformation applications while protecting the privacy and integrity of data
- **Data Consumer**: Includes end users or other systems that use the results of the Big Data Application Provider

정유산업 하둡 아키텍쳐

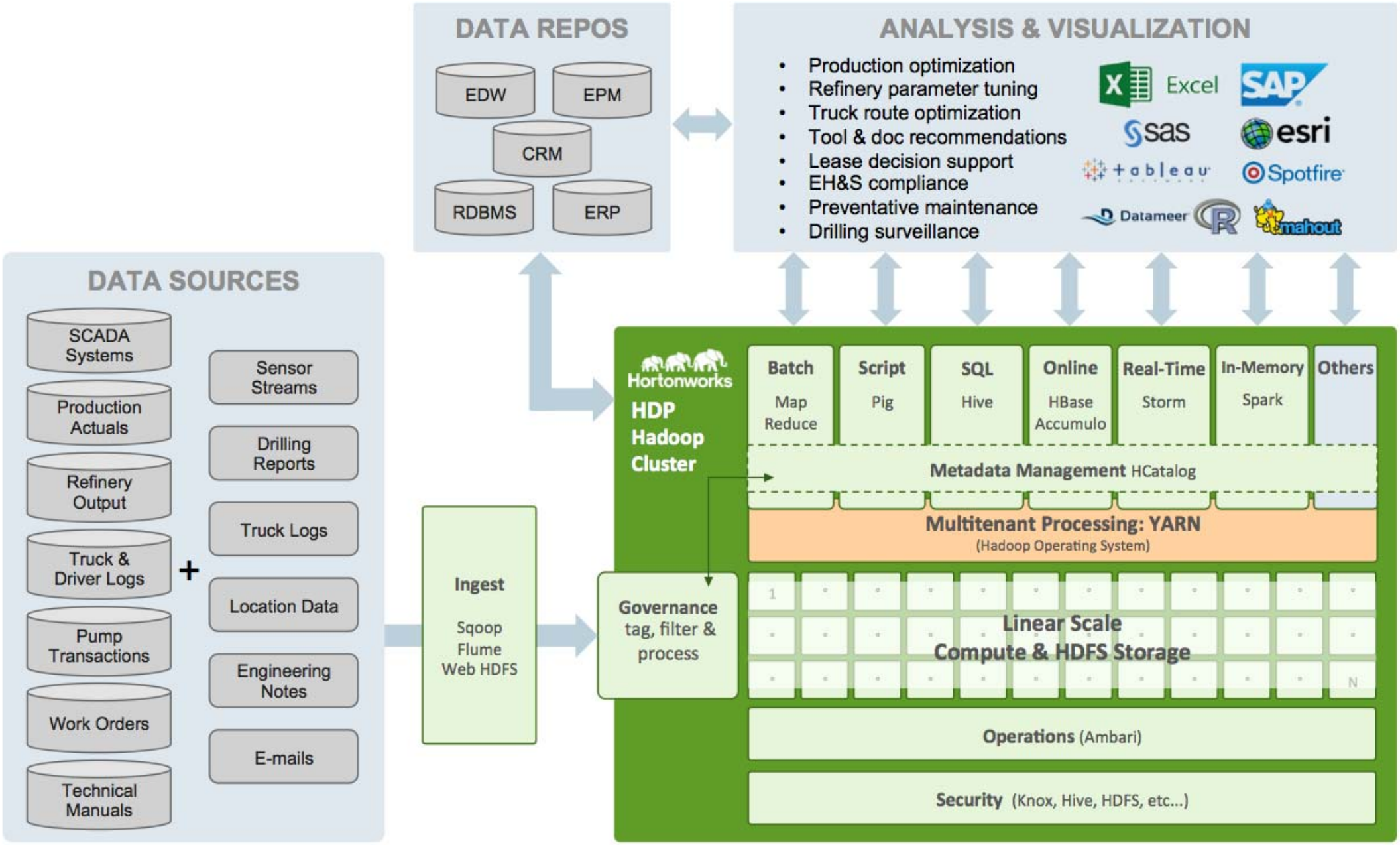
❑ 비즈니스 요구사항

- 새로운 장치들의 도입
- 프로세스 자동화
- 다양한 조직간의 협업의 증대

❑ 다양하고 방대한 데이터 소스

- 센서, 지리, 날씨 정보
- 방대한 채굴 정보
- 지진계측 데이터

정유산업 하둡 아키텍쳐

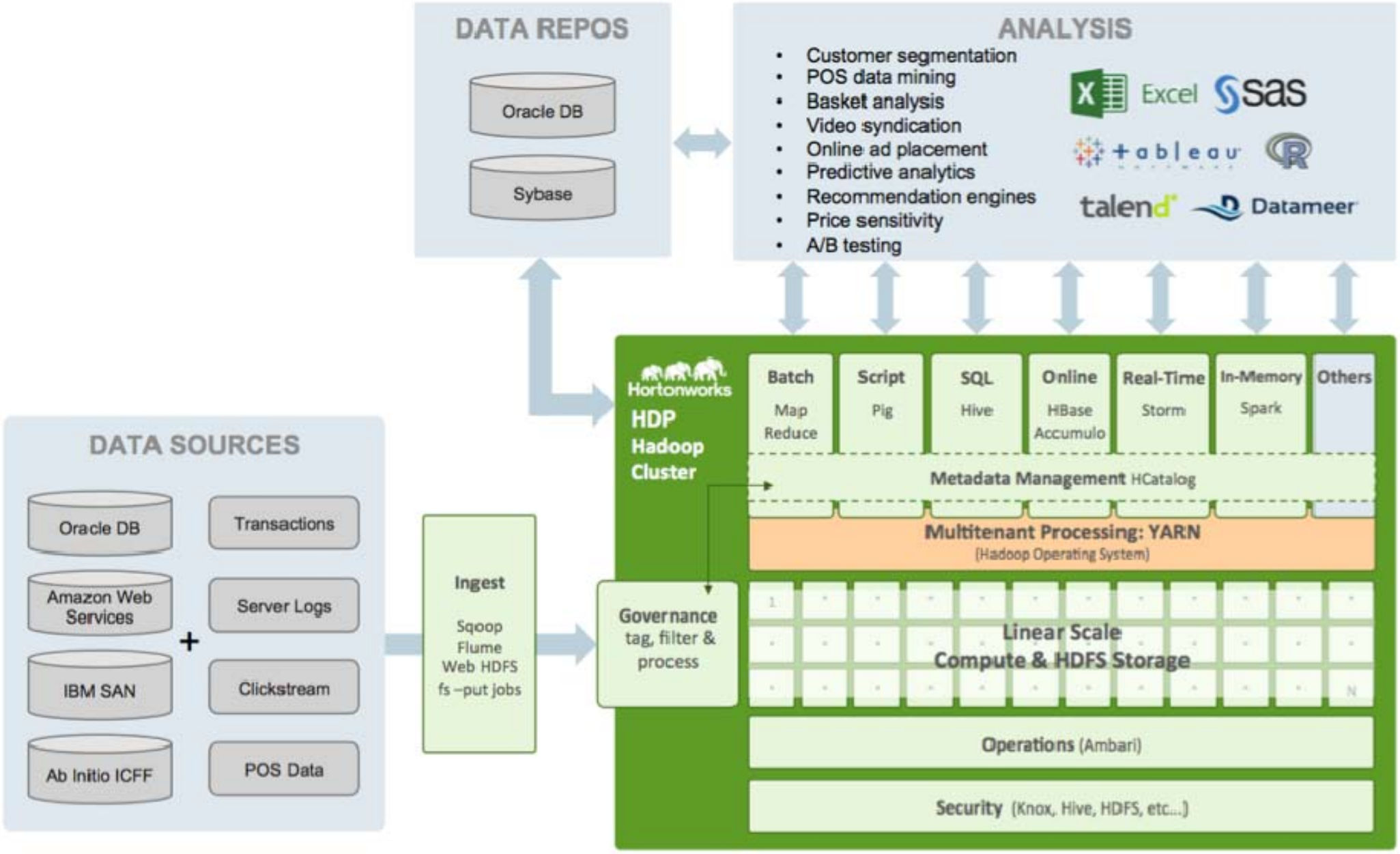

DATA REPOS
EDW
EPM
CRM
RDBMS
ERP
ANALYSIS & VISUALIZATION
Production optimization
Refinery parameter tuning
Truck route optimization
Tool & doc recommendations
Lease decision support
EH&S compliance
Preventative maintenance
Drilling surveillance
Excel
SAP
sas
esri
tableau
Spotfire
Datameer
mahout
DATA SOURCES
SCADA Systems
Production Actuals
Refinery Output
Truck & Driver Logs
Pump Transactions
Work Orders
Technical Manuals
+
Sensor Streams
Drilling Reports
Truck Logs
Location Data
Engineering Notes
E-mails
Ingest
Sqoop
Flume
Web HDFS
Hortonworks
HDP
Hadoop
Cluster
Batch
Map Reduce
Script
Pig
SQL
Hive
Online
HBase
Accumulo
Real-Time
Storm
In-Memory
Spark
Others
Metadata Management HCatalog
Multitenant Processing: YARN
(Hadoop Operating System)
Governance
tag, filter & process
Linear Scale
Compute & HDFS Storage
Operations (Ambari)
Security (Knox, Hive, HDFS, etc...)

광고산업 하둡 아키텍쳐 활용 예

❑ POS 데이터 분석을 통해서 충성도가 높은 고객을 식별하기

- 사실상 고객의 구매 패턴을 파악할 수 있는 가장 좋은 기반 데이터
- 데이터의 양과 스트리밍 속도라는 두 가지 측면에서 아직 이 데이터를 가지고 분석을 하는 단계에는 도달하지 못함
- 대량의 스트리밍 데이터를 합리적인 비용으로 처리하기 위한 아키텍쳐를 제공
- 다양한 분석 – 고객의 구매 패턴, 가격에 대한 민감성(Sensitivity) 수요 예측(demand forecasting)
- 최근의 Stinger와 같은 도구를 사용하여 스트리밍 데이터에 대해 신속한 분석 계획

광고산업 하둡 아키텍쳐



광고산업 하둡 아키텍쳐 활용 예

❑ LUMINA

- 라틴계 고객들에 대한 타겟 광고를 전문적으로 진행하는 회사
- 미국에 살고 있는 라틴계 소비자들의 샘플을 통해서 인사이트를 얻었음
- 미국에 살고 있는 전 라틴계 소비자들로 확대하여 실증적인 근거 데이터를 확보하는 것이 목표
- 호튼웍스는 기존의 Luminar가 가진 BI 시스템(AWS, R, Tableau 등)에 대한 투자를 그대로 보존하면서 기존의 300개의 데이터소스에서 하둡을 통해 2000개 이상의 데이터소스를 처리할 수 있도록 구축함으로써 샘플링 기법이 아닌 전수 조사를 통한 타겟 광고 시스템을 구축
- 현재 한 달에 15TB에 해당하는 데이터 처리

기업의 빅데이터 활용 목적

❑ **(Gain Insight)**

- 기업 경영에 새로운 통찰력을 얻고자 함

❑ **(Take Action)**

- 경영 환경에 대한 분석과 통찰력을 통해 실행하는 기업 환경을 조성하고자 함.

❑ **(See Everything)**

- 데이터를 단순히 분석하는데 그치는 것이 아니라 분석한 데이터를 이해하고 공격적으로 실행하는 기업 환경의 변화는 실제 기업 내에 모든 데이터를 관리하고 모니터링

❑ **(Dark Data)**

- 감춰져 있거나 드러나지 않은 데이터를 분석 범주 내에 포함시켜서 시스템 안으로 끌어들여 공격적인 마케팅을 구현하고자 함

❑ **(Miss Nothing)**

- 경영에 있어서 위험 요인을 줄이고 실수를 반복하지 않으려는 의도

❖ 구글은 빅데이터를 어떻게 활용했는가? **“People Analytics” by Ben Waber**

지난주 과제

❑ AWS

- Amazon Elastic MapReduce(EMR)
- Amazon Redshift
- Amazon Kinesis Firehose
- AWS IoT, Amazon Machine Learning ...

하둡 및 Spark

Amazon EMR

몇 분 만에 완전관리형 하둡 프레임워크를 손쉽게 프로비저닝할 수 있습니다. 하둡 클러스터를 동적으로 확장하고 사용한 만큼만 비용을 지불하십시오. Apache Spark 및 Presto와 같은 인기 있는 분산 프레임워크를 실행할 수 있습니다. 자세히 알아보기 »

euclid

Retail and POS Analytics

Process 10's of TB in hours vs. 2 weeks

80-90% reduction in costs