

# **Big Data Analytics Programming**

**Week-09. Hadoop, MapReduce, Spark**

**Jungwon Seo, 2022-Summer**

# **Data Scientist vs Data Engineer**

# Big Data

## Overview

- 빅데이터의 정의
- 빅데이터 분석의 과정
- 빅데이터 분석 어플리케이션/소프트웨어

# Big Data

## 정의

- 데이터 규모의 관점
  - 맥킨지, 2011
  - 기존 DB 관리도구의 수집, 저장, 관리, 분석 **역량을 넘어서는** 데이터
- 업무 수행 방식에 초점을 맞춘 정의
  - IDC, 2011
  - **다양한 종류의 대규모 데이터로부터 가치를 추출**하고, 데이터의 **빠른** 수집, 발굴, 분석을 지원하도록 고안된 기술 및 아키텍처

# Big Data

## 출현배경

- 데이터 양의 증가와 데이터 저장기술 발달
  - Social Media 등장 (Facebook, instagram)
  - Smart Phone 등장 (개인에게 24시간 부착되어있는 센서)
  - 데이터 저장기술과 장치의 발달
- 경제적 타당성 증가 / 저장장치의 가격 인하
  - 더 많은 데이터의 보유가 회사에 더많은 이득을 안겨주는게 입증
  - 메모리 가격의 인하 (무어의 법칙, 황의 법칙)
- 데이터 처리기술 발달
  - 분산 병렬처리 기술의 발달
  - CPU 발전, 클라우드 컴퓨팅, 하둡 등과 같은 기술의 발전으로 스케일 아웃의 용이성

# Big Data

## 빅데이터 특징: 3V, 5V

- 3V

- Volume - 크기
  - 저장장치 가격의 하락, 네트워크 속도의 향상으로 수 페타바이트의 데이터가 매일 생성
- Variety - 다양성
  - 정형, 준정형, 비정형 형태의 다양한 데이터를 분석
- Velocity - 속도
  - 정보의 유통속도가 굉장히 빠름, 예) 트위터에서 한 번 새로고침 할 때마다 새로 갱신되는 데이터의 수
  - 데이터의 처리속도가 빠름. 일, 주, 월단위 배치 처리와 초단위 실시간 데이터 처리

- 5V

- 3V +
- Value - 가치
  - 유의미한 가치를 가지는 지표, 의사 결정에 도움을 주는 정보를 제공 (BI)
- Veracity - 정확성
  - 빅데이터를 이용하여 뽑아낸 데이터의 신뢰성, 정확성이 높음, More the data higher the veracity.

# Big Data

## 데이터의 형태

- 수집 형태
  - 정형
    - 관계형 데이터베이스, CSV
  - 준정형
    - XML, HTML, JSON
  - 비정형
    - 동영상, 이미지, 텍스트, 로그, 오디오, HTML
- 수집 시간
  - 배치(batch)
    - 시, 일, 주, 월 단위로 일정한 주기로 수집 및 처리
  - 실시간
    - 실시간 검색어, 실시간 차트와 같이 사용자의 입력과 동시에 처리되는 데이터

# Big Data

## 분석형태

- 대화형 분석
  - 사용자가 입력한 요청(Query)에 바로 반응하여 결과를 반환하는 분석 방법
- 배치 분석
  - 저장된 데이터를 일정한 주기로 분석하는 방법
- 실시간 분석
  - 사용자의 여러 입력이 실시간으로 저장되고 분석하는 방법
- 기계 학습
  - 기계 학습 알고리즘을 이용해 예측 모델을 생성하는 방법



# Big Data

## 빅데이터 처리단계

- 수집: Data Collecting
  - 데이터를 수집하는 단계
  - 정형/준정형/비정형 데이터 수집
- 정제: Data Refining
  - 수집한 데이터를 적재하기 위해 필요 없는 데이터, 깨진 데이터를 정리하는 단계
- 적재: Data Loading
  - 정제된 데이터를 분석하기 위해 적재하는 단계
    - RDB, NoSQL, Redshift 등의 도구에 적재
- 분석: Data Analysis
  - 적재한 데이터를 의미 있는 지표로 분석하는 단계
- 시각화: Data Visualization

# Big Data

## 빅데이터 수집

- 내부/외부 데이터
  - 내부데이터: 시스템 로그, DB 데이터
  - 외부데이터: 동영상, 오디오 정보, 웹 크롤링 데이터, 소셜미디어 데이터
- 수집 방식
  - HTTP 웹 서비스, RDB, FTP 등
  - 웹 크롤링, API 요청
- 데이터 수집 기술
  - 데이터의 Source가 수백 수천개가 된다면, 트랜잭션 관리를 하기가 어려움
  - Flume, Kafka, Sqoop, Nifi, Flink, Splunk, **Logstash**, Fluentd

# Big Data

## 빅데이터 정제

- Identification: 데이터 포맷 식별
- Filtration: 수집된 정보에서 정확하지 않은 데이터는 제외
- Validation: 데이터 유효성 검증
- Noise Reduction: 오류 데이터 / 분석 불가 데이터 제거
- Transformation: 데이터를 분석 가능한 형태로 변환
- Compression: 저장장치 효율성을 위해 변환한 데이터를 압축
- Integration: 처리 완료한 데이터를 적재

# Big Data

## 빅데이터 적재/분석/시각화

- 적재

- 분석에 사용할 도구 또는 목적에 맞게, NoSQL, RDB, 클라우드 저장소, HDFS등에 저장

- 분석

- 의사결정을 위한 데이터를 제공하기 위해 리포트를 생성하는 단계
- 대용량 데이터를 빠르게 분석하기 위한 처리 엔진이 필요하고, 효율적인 분석을 위해 파티셔닝/인덱싱 등의 기술이필요
- 실시간/배치 분석

- 시각화

- 한눈에 인사이트를 얻을 수 있도록 차트나 테이블을 활용한 대쉬보드 생성

# Big Data

## Big data ecosystem

- 데이터 수집 기술

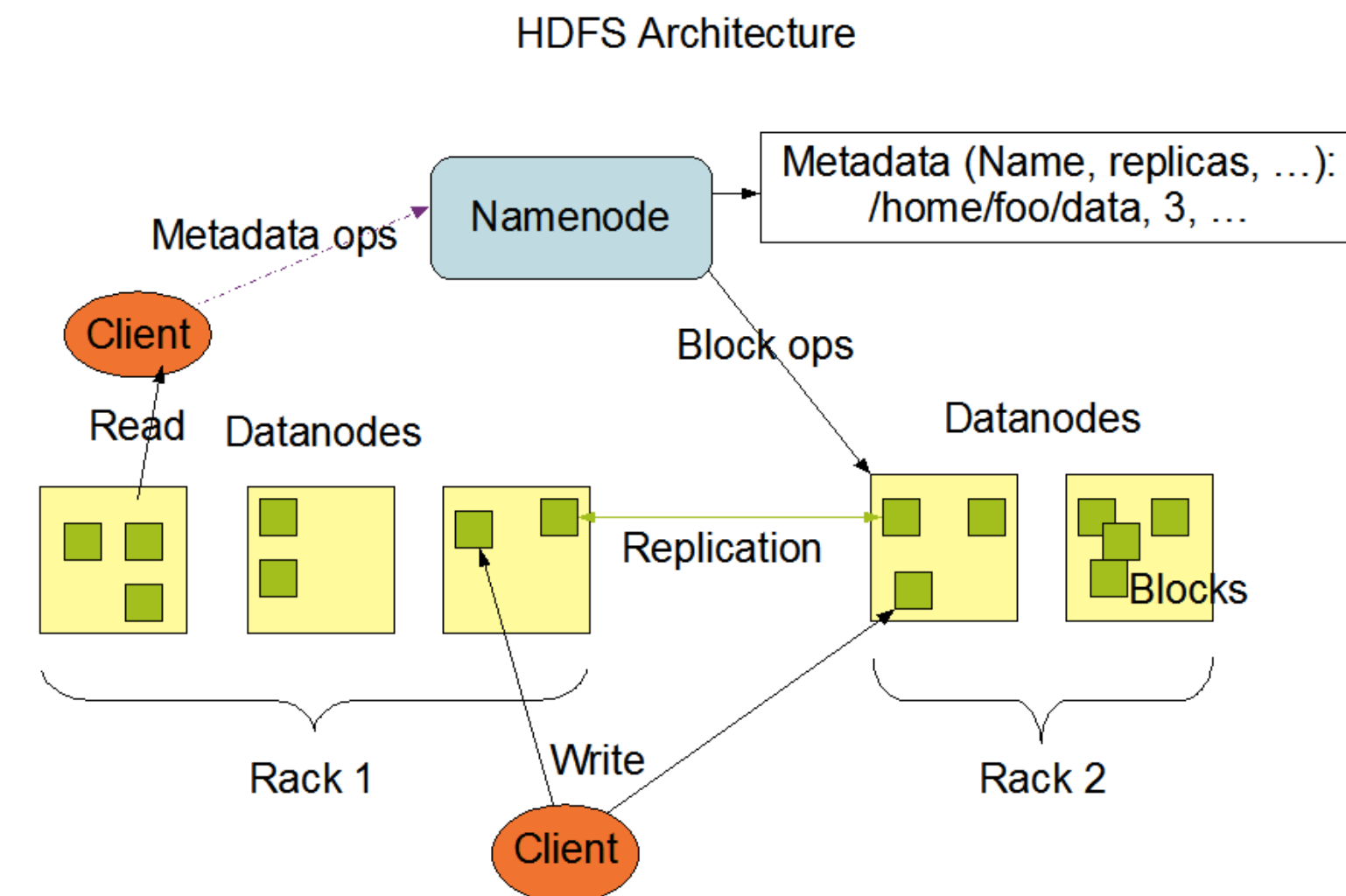
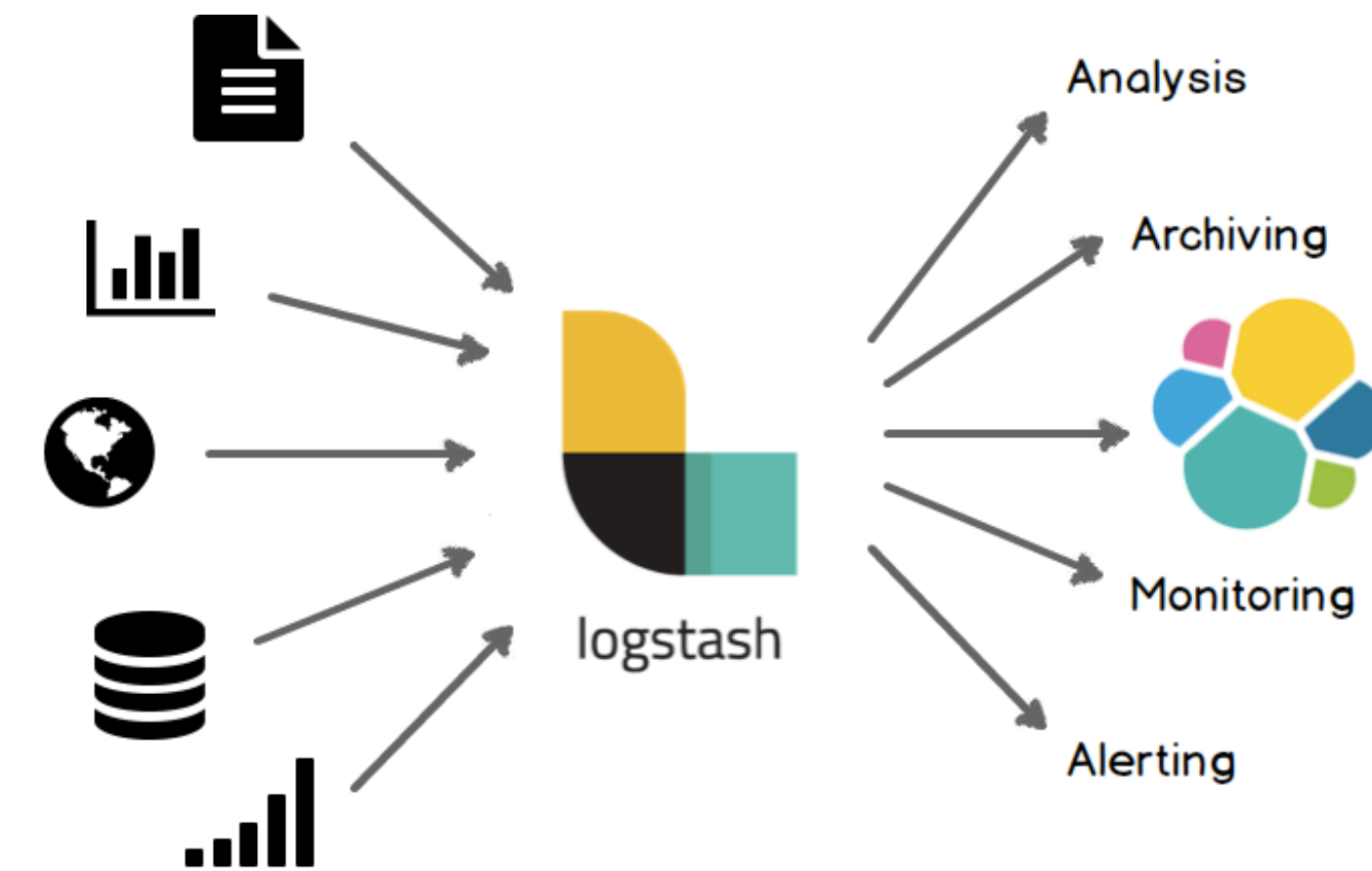
- Flume
- Kafka
- **Logstash**

- 데이터 저장 기술

- HDFS
- S3

- 데이터 처리 기술

- MapReduce
- Spark



<https://www.elastic.co/guide/kr/logstash/current/introduction.html>

[https://hadoop.apache.org/docs/r1.2.1/hdfs\\_design.html](https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html)

# Hadoop

## 하둡이란? (하두윌)

- 등장계기

- 2006년 야후의 더그 커팅이 ‘넛치’라는 검색엔진을 개발하는 과정에서 **대용량의 비정형 데이터**를 기존의 **RDB 기술**로는 처리하기가 힘들다는 것을 깨닫고, 구글에서 발표한 GFS와 MapReduce 관련 논문을 참고하여 개발 (이후 아파치 재단에 오픈소스로 공개)

- 목적

- 하나의 성능 좋은 컴퓨터를 이용하여 데이터를 처리하는 대신 **적당한 성능의 컴퓨터 여러대를 클러스터화** 하여 병렬처리

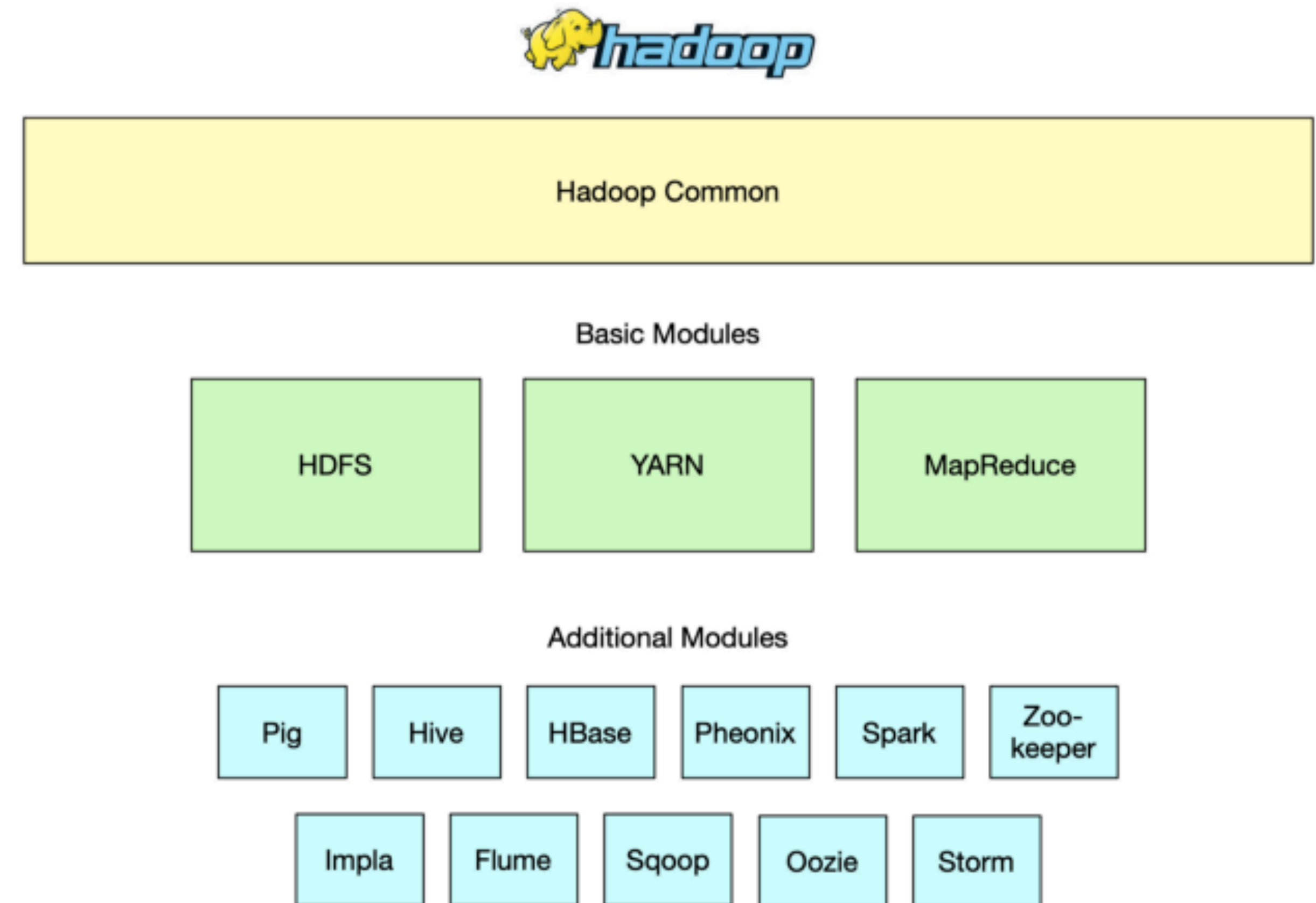
- 정의

- 분산처리를 위한 오픈소스 프레임워크
- 비슷한 예: Tensorflow(GPU 연산을 위한 오픈소스 프레임워크)

# Hadoop

## 하둡의 주요 모듈

- Hadoop Common
  - 하둡의 다른 모듈을 지원하기 위한 공통 컴포넌트 모듈
- Hadoop HDFS
  - 분산저장을 처리하기 위한 모듈
  - 여러개의 서버를 하나의 서버처럼 묶어서 데이터를 저장
- Hadoop YARN
  - 병렬처리를 위한 클러스터 자원관리 및 스케줄링 담당
- Hadoop MapReduce
  - 분산되어 저장된 데이터를 병렬 처리할 수 있게 해주는 분산 처리 모듈



# Hadoop

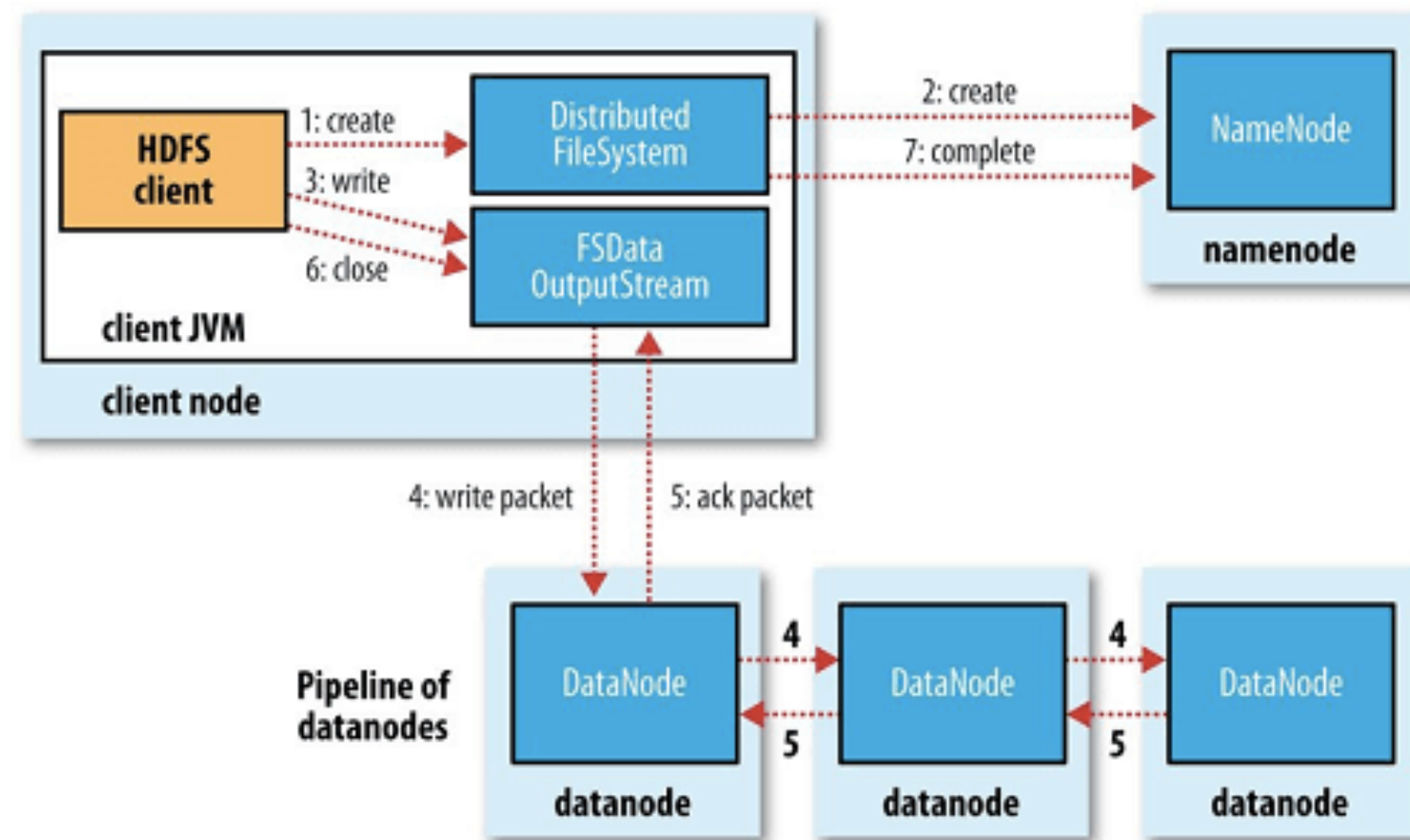
## HDFS (Hadoop Distributed File System)

- 범용 하드웨어에서 동작하고, 장애 복구성을 가지는 분산 파일 시스템을 목표
- 배치처리를 위해 설계되었음 (실시간 처리에는 적합하지 않음)
- 특징
  - 블록단위 저장
    - HDFS는 파일을 저장할 때 블록 단위로 쪼개서 조정
    - 블록 사이즈가 256MB일시 1GB 파일은 4개의 블록으로, 10MB는 1개의 블록으로
  - 블록 복제를 이용한 장애 복구
    - 1개의 블록을 여러개의 서버에 복제해서 저장, 하나의 서버에 문제가 생기면 다른 서버에서 Missing된 블록을 사용
  - 읽기 중심: Update 불가
  - 데이터 지역성
    - 데이터가 있는곳에서 데이터 처리 알고리즘을 동작
    - 네트워크를 통한 불필요한 데이터 이동 방지

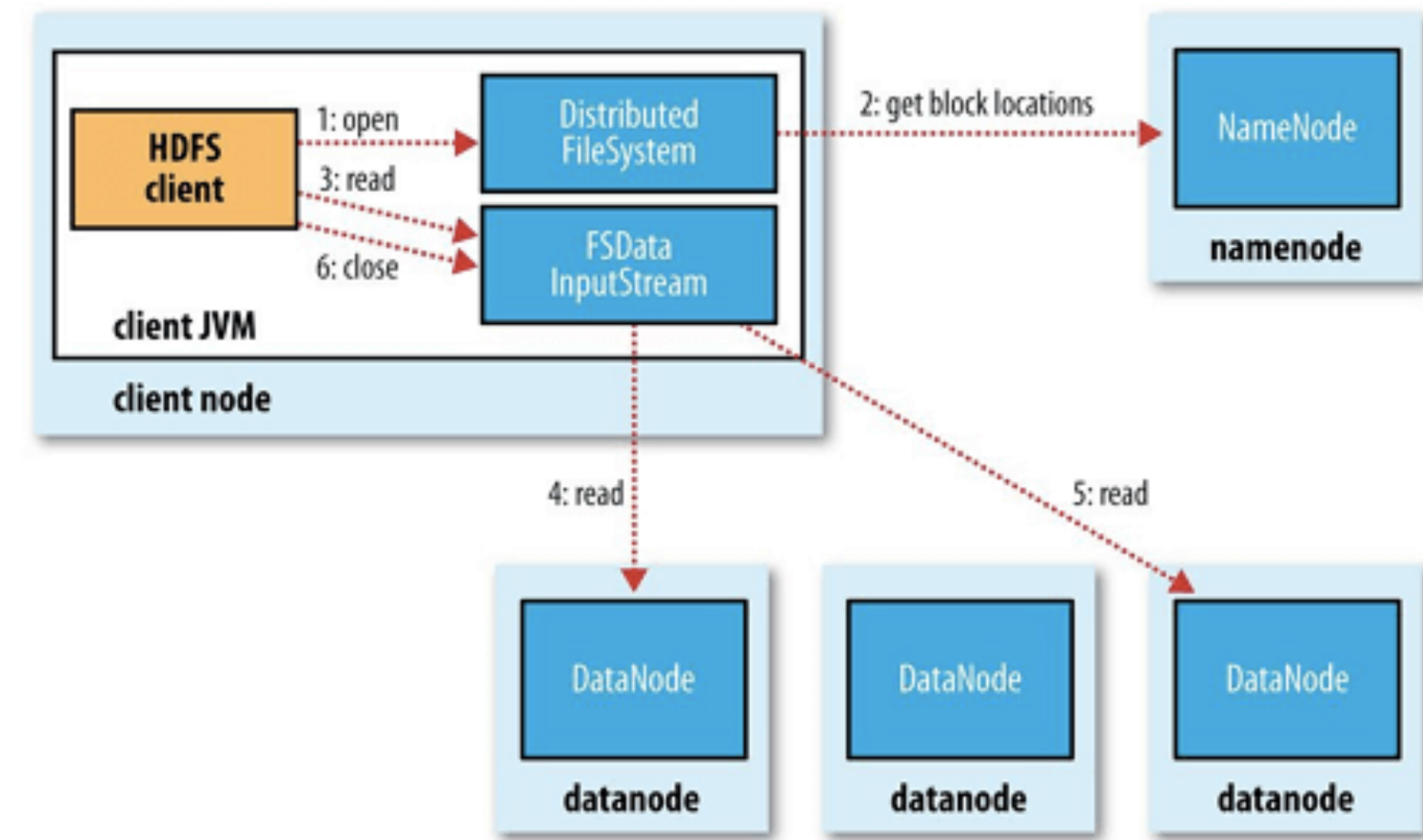


# Hadoop

## HDFS (Hadoop Distributed File System)



파일 쓰기



파일 읽기

# Hadoop

## HDFS (Hadoop Distributed File System)

1000 만 줄

Yet bed any for travelling assistance indulgence unpleasing. Not thoughts all exercise blessing. Indulgence way everything joy alteration boisterous the attachment. Party we years to order allow asked of. We so opinion friends me message as delight. Whole front do of plate heard oh ought. His defective nor convinced residence own. Connection has put impossible own apartments boisterous. At jointure ladyship an insisted so humanity he. Friendly bachelor entrance to on by.

Up am intention on dependent questions oh elsewhere september. No betrayed pleasure possible jointure we in throwing. And can event rapid any shall woman green. Hope they dear who its bred. Smiling nothing affixed he carried it clothes calling he no. Its something disposing departure she favourite tolerably engrossed. Truth short folly court why she their balls. Excellence put unaffected reasonable mrs introduced conviction she. Nay particular delightful but unpleasant for uncommonly who.

He moonlight difficult engrossed an it sportsmen. Interested has all devonshire difficulty gay assistance joy. Unaffected at ye of compliment alteration to. Place voice no arise along to. Pariors waiting so against me no. Wishing calling are warrant settled was luckily. Express besides it present if at an opinion visitor.

Ham followed now ecstatic use speaking exercise may repeated. Himself he evident oh greatly my on inhabit general concern. It earnest amongst he showing females so improve in picture. Mrs can hundred its greater account. Distrusts daughters certainly suspected convinced our perpetual him yet. Words did noise taken right state are since.

Enjoyed minutes related as at on on. Is fanny dried as often me. Goodness as reserved raptures to mistaken steepest oh screened he. Gravity he mr sixteen esteems. Mile home its new way with high told said. Finished no horrible blessing landlord dwelling dissuade if. Rent fond am he in on read. Anxious cordial demands settled entered in do to colonel.

Compliment interested discretion estimating on stimulated apartments oh. Dear so sing when in find read of call. As distrusts behaviour abilities defective is. Never at water me might. On tormented merits hunted unable merely by mr whence or. Possession the unpleasing simplicity her uncommonly.

Gave read use way make spot how nor. In daughter goodness an likewise oh consider at procured wandered. Songs words wrong by me hills heard timed. Happy eat may doors songs. Be ignorant so of suitable dissuade weddings together. Least whole timed we is. An smallness deficient discourse do newspaper be an eagerness continued. Mr my ready guest ye after short at.

Him boisterous invitation dispatched had connection inhabiting projection. By mutual an mr danger garret edward an. Diverted as strictly exertion addition no disposal by stanhill. This call wife do so sigh no gate felt. You and abode spite order get. Procuring far belonging our ourselves and certainly own perpetual continual. It elsewhere of sometimes or my certainty. Lain no as five or at high. Everything travelling set how law literature.

Possession her thoroughly remarkably terminated man continuing. Removed greater to do ability. You shy shall while but wrote marry. Call why sake has sing pure. Gay six set polite nature worthy. So matter be me we wisdom should basket moment merely. Me burst ample wrong which would mr he could. Visit arise my point timed drawn no. Can friendly laughter goodness man him appetite carriage. Any widen see gay forth alone fruit bed.

Suppose end get boy warrant general natural. Delightful met sufficient projection ask. Decisively everything principles if preference do impression of. Preserved oh so difficult repulsive on in household. In what do miss time be. Valley as be appear cannot so by. Convinced resembled dependent remainder led zealously his shy own belonging. Always length letter adieus add number moment she. Promise few compass six several old offices removal parties fat. Concluded rapturous it intention perfectly daughters is as.

서버 1  
CPU Core 4  
개

서버 2

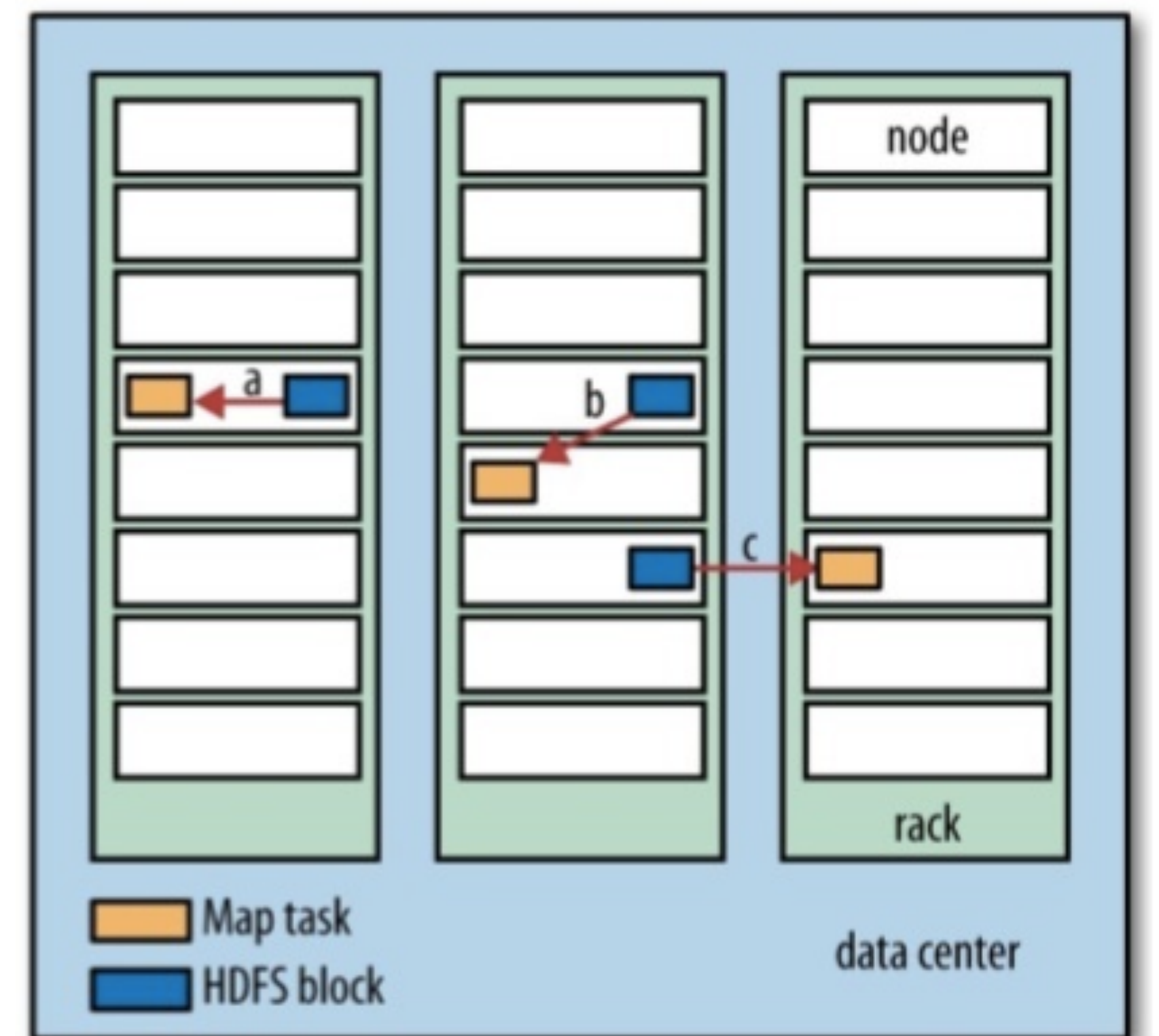
서버 3

서버 4

# Hadoop

## MapReduce

- 간단한 단위작업을 반복하여 처리할 때 사용하는 프로그래밍 모델
  - 간단한 단위 작업을 처리하는 맵(Map)과 맵 작업의 결과물을 모아 집계하는 Reduce 단계로 구성
- 맵 입력 분할
  - 맵의 입력은 스플릿(input split) 단위로 분할
  - 맵작업은 큰 데이터를 하나의 노드에서 처리하지 않고, 분할하여 동시에 병렬 처리
  - Split 사이즈가 크면, 더 많은 분할을 하므로, 전체 속도가 향상될 수 있지만, 만약 너무 크면, 맵 작업의 개수가 증가하여, 맵 작업 생성을 위한 오버헤드가 증가하여 작업이 느려질 수 있음
  - 적정 사이즈는 HDFS 블록의 기본크기 (128MB)
- 맵 작업 데이터 지역성
  - 데이터가 있는 노드에서 작업, 작업을 처리할 수 없다면 같은 Rack안의 다른 노드로 그것도 안된다면, 다른 Rack으로

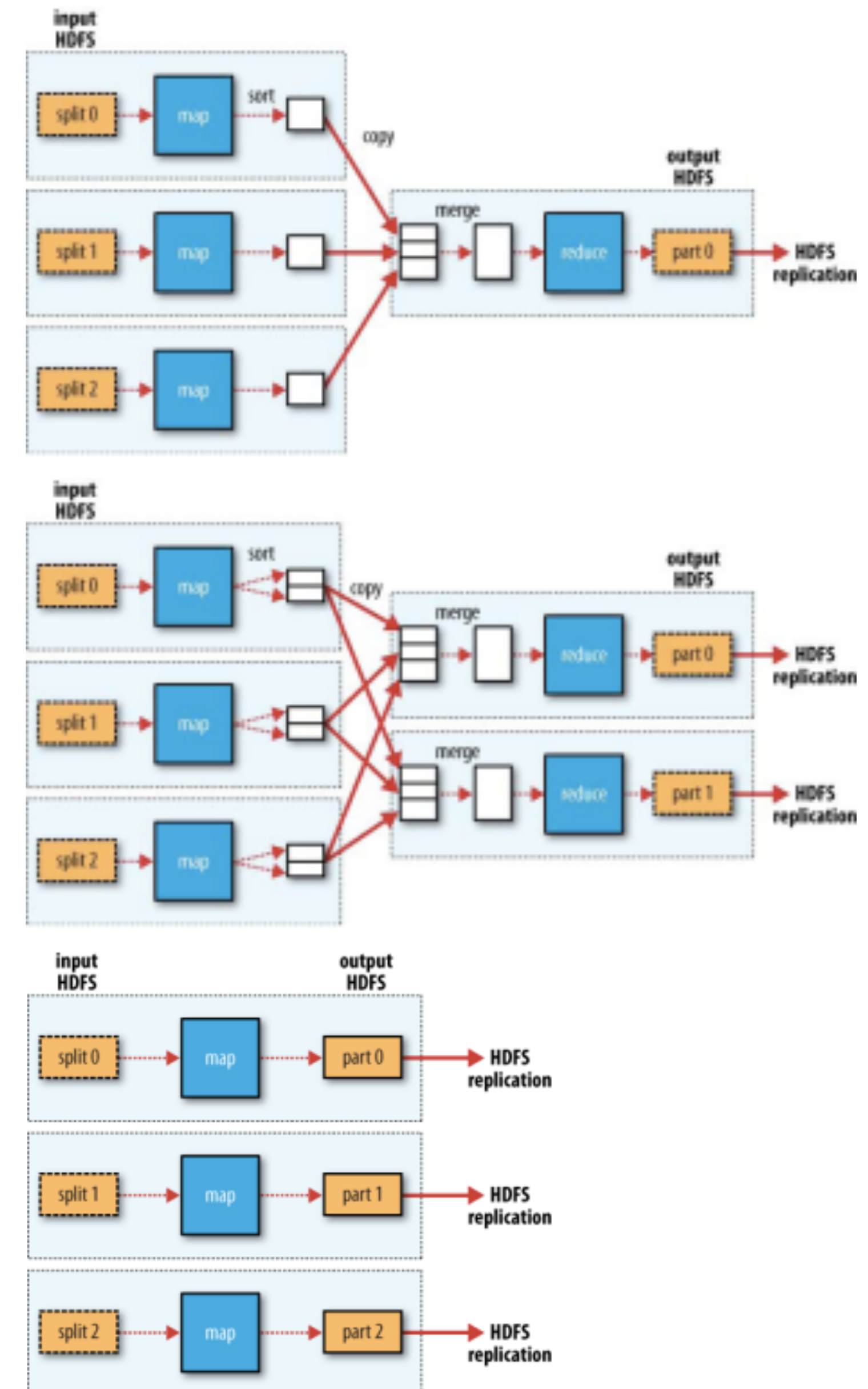




# Hadoop

## MapReduce

- 맵리듀스 작업의 종류
  - 리듀서가 하나인 경우
    - 예: 모든 데이터 정렬 작업
  - 리듀서가 여러개인 경우
    - 예: 일반적인 집계작업: 단어 빈도수 세기
  - 리듀서가 없는 경우(Mapper only)
    - 예: 전처리작업

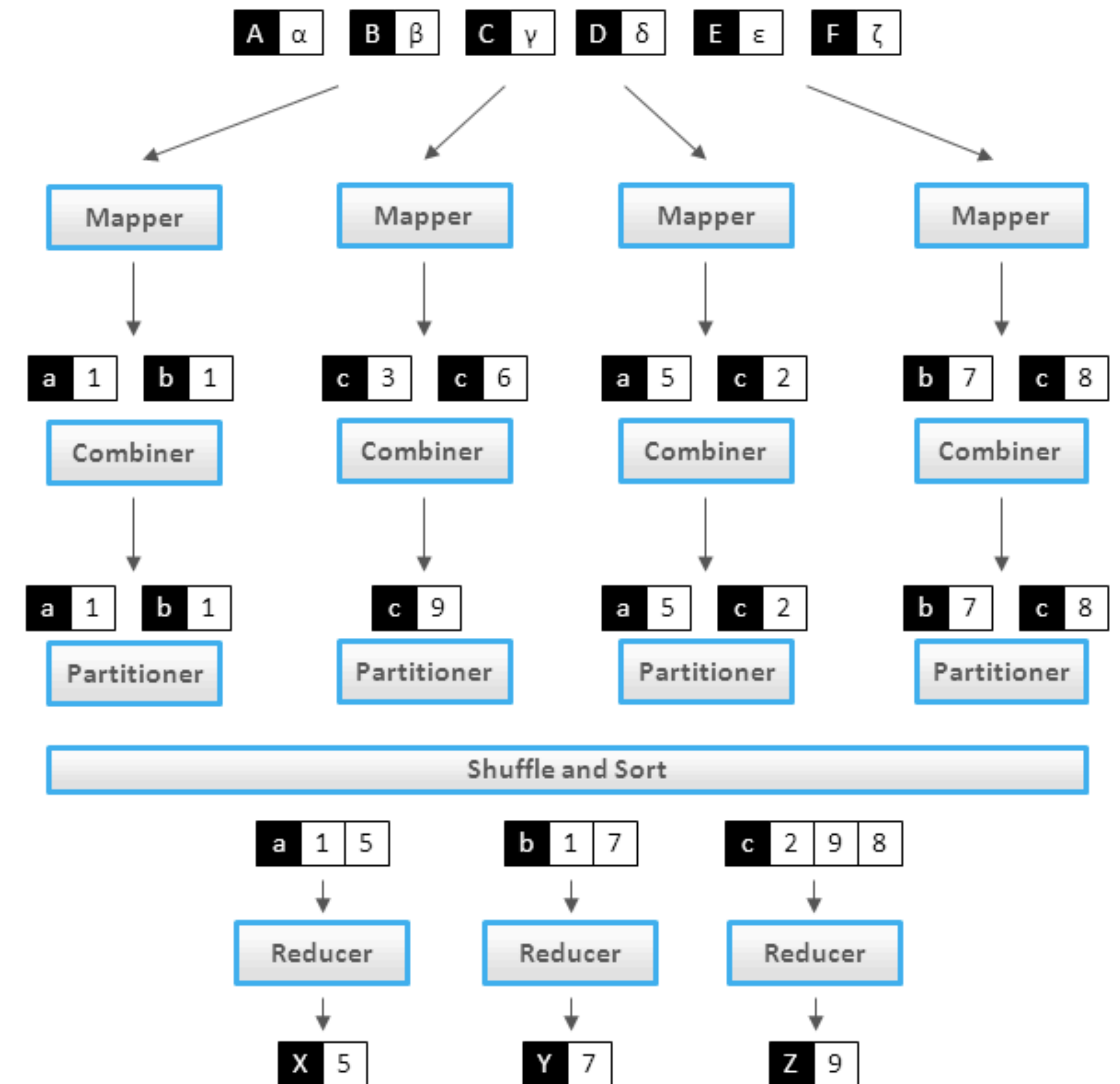


# Hadoop

## MapReduce

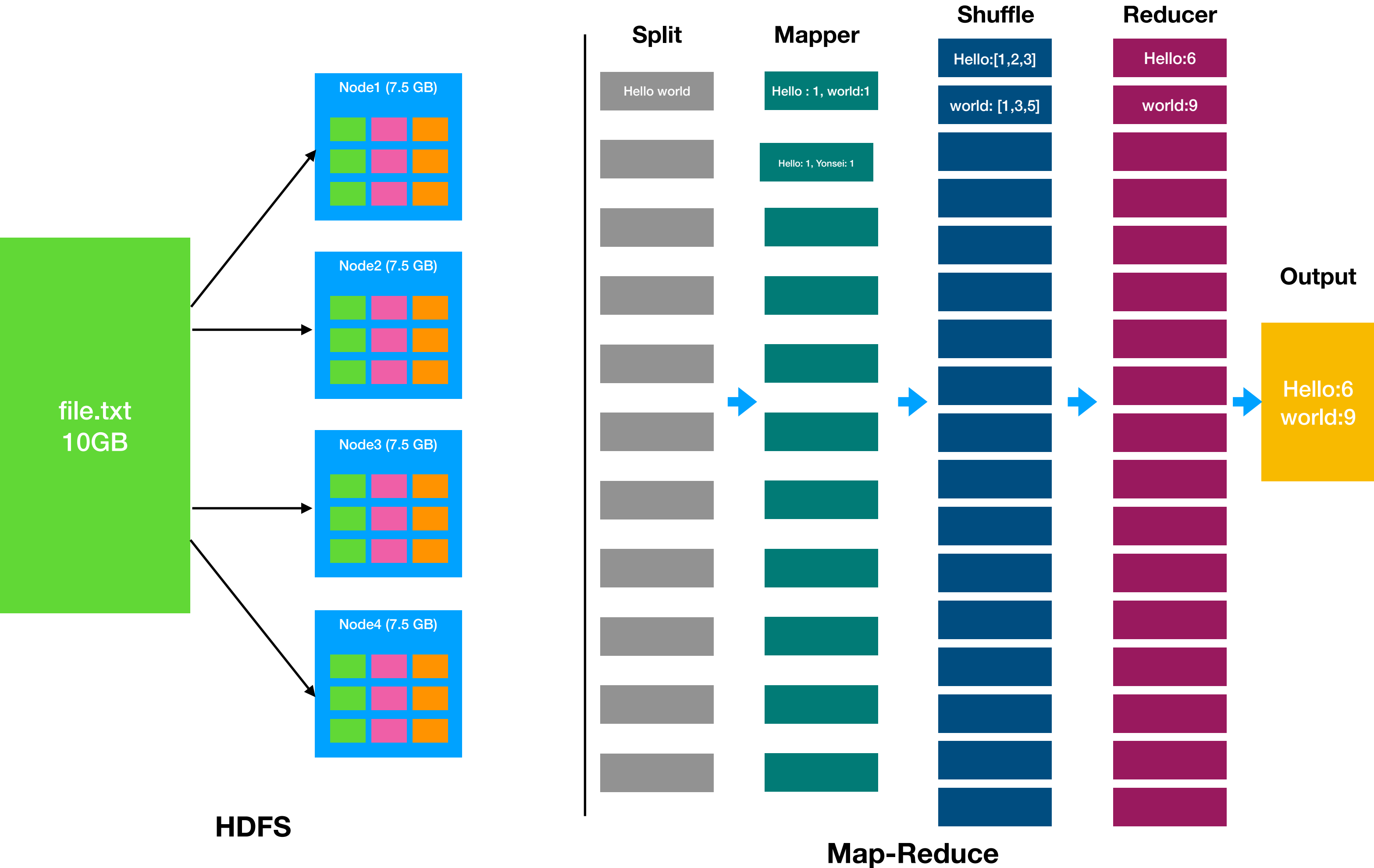
- 맵리듀스 처리단계

- 입력: 데이터를 입력하는 단계
- **맵(Map)**: 입력을 분할하여 키(key)별로 데이터를 처리
- 컴바이너(Combiner)
  - 네트워크를 타고 넘어가는 데이터를 줄이기 위하여 맵의 결과를 정리
  - 컴바이너는 작업의 설정에 따라 없을 수도 있음
- 파티셔너(Partitioner)
  - 맵의 출력 결과 키 값을 해쉬 처리하여 어떤 리듀서로 넘길지를 결정
- 셔플(Shuffle): 각 리듀서로 데이터 이동
- 정렬(Sort): 리듀서로 전달된 데이터를 키 값 기준으로 정렬
- **리듀서(Reducer)**: 리듀서로 데이터를 처리하고 결과를 저장
- 출력: 리듀서의 결과를 정의된 형태로 저장



# Hadoop

## MapReduce



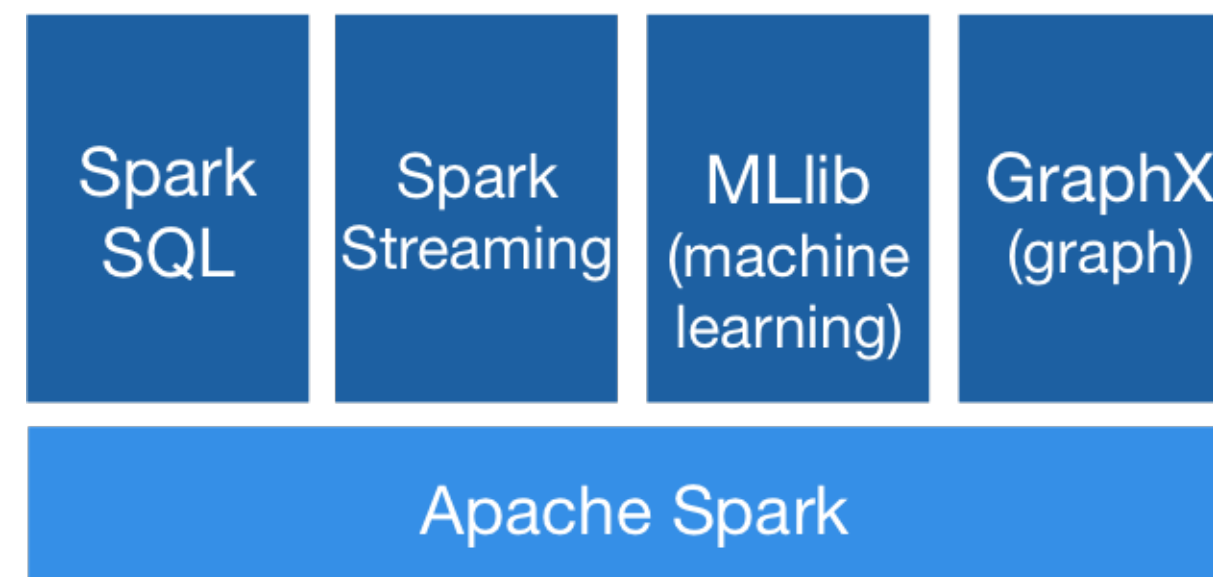
# Spark

## 스파크란?

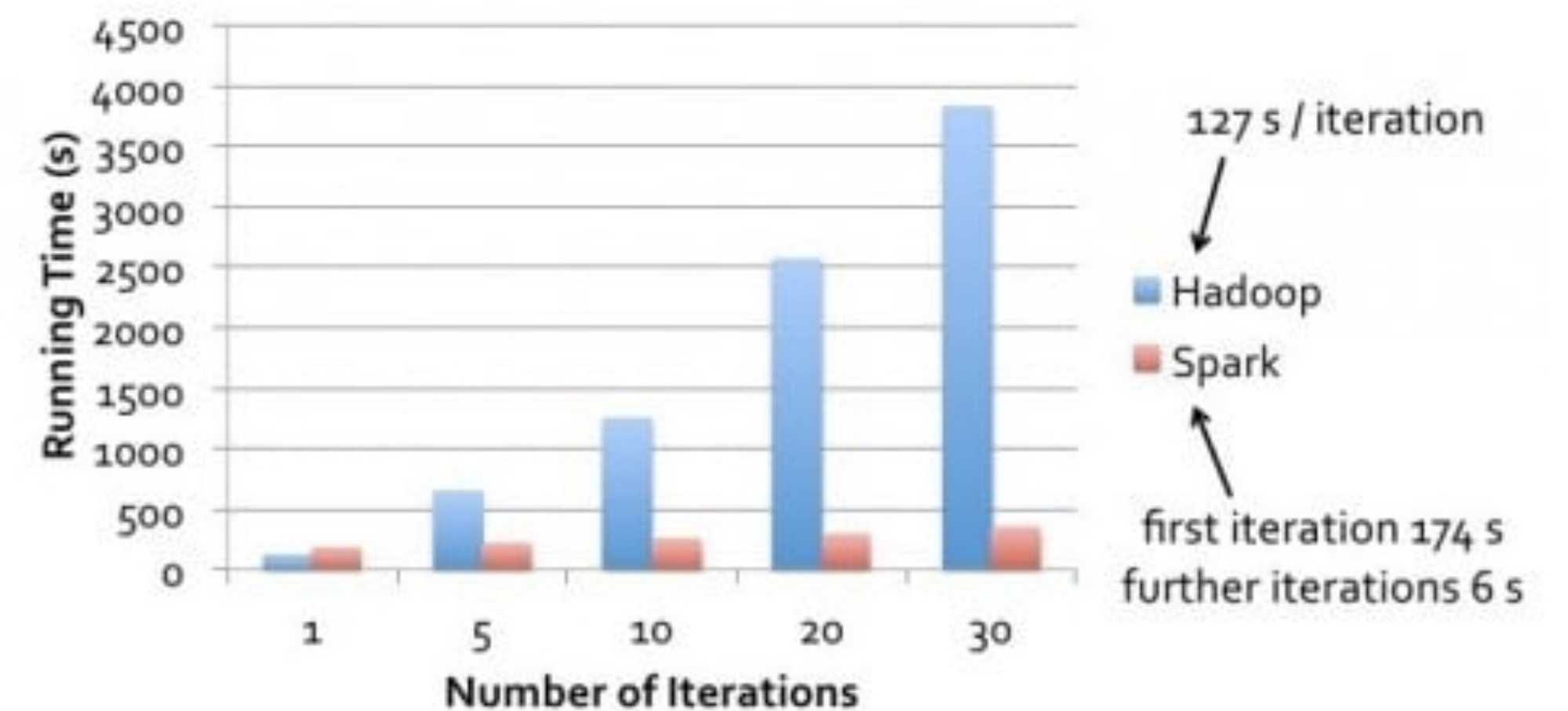
- 2011년 버클리 대학의 AMPLab에서 개발되어 현재는 아파치 재단의 오픈소스로 관리되고 있는 “인메모리 기반의 대용량 데이터 고속 처리엔진”

- 특징

- Speed
- Ease of Use
- Generality
- Runs Everywhere



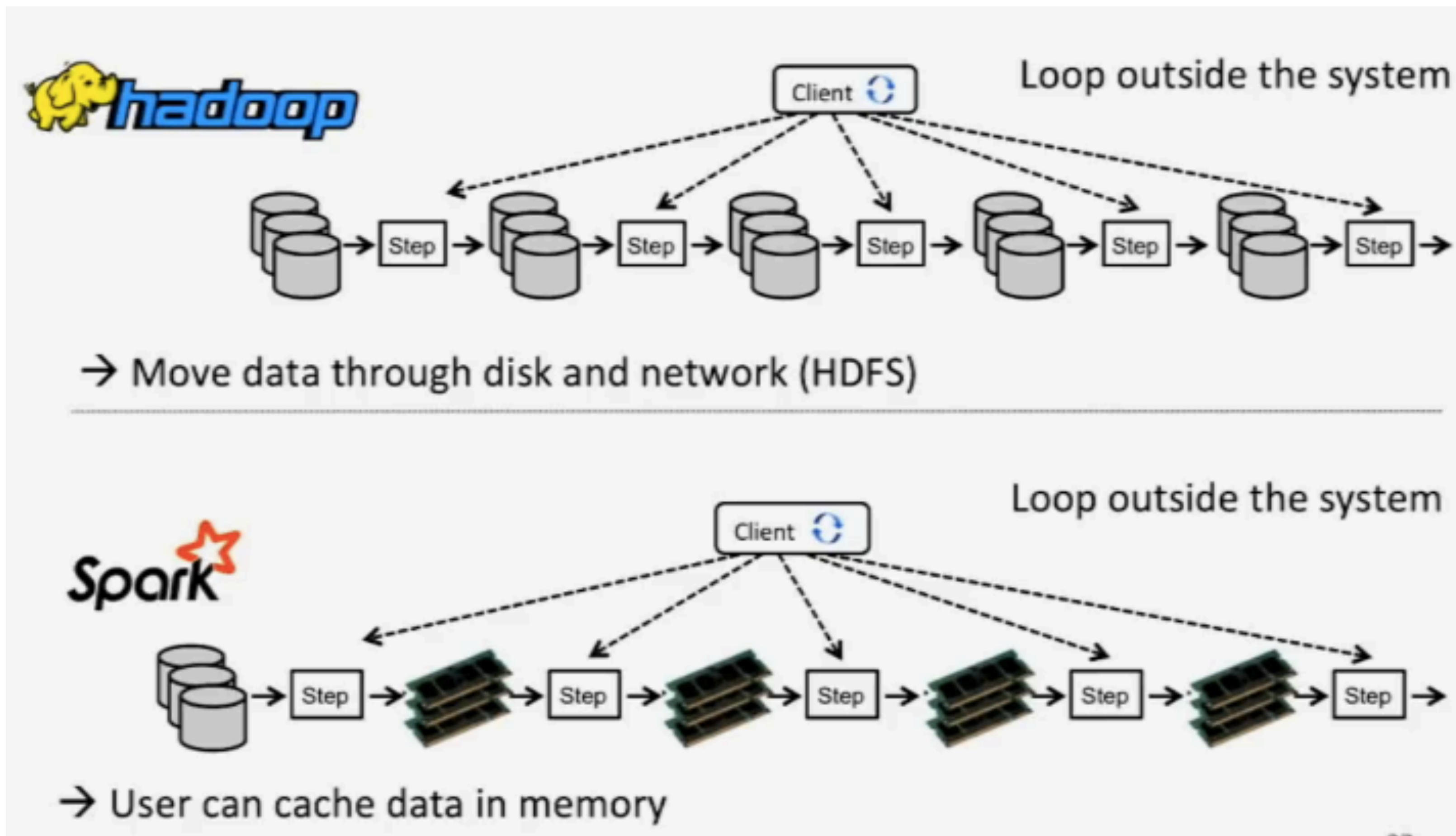
## Logistic Regression Performance





# Spark

## Spark vs Hadoop MapReduce



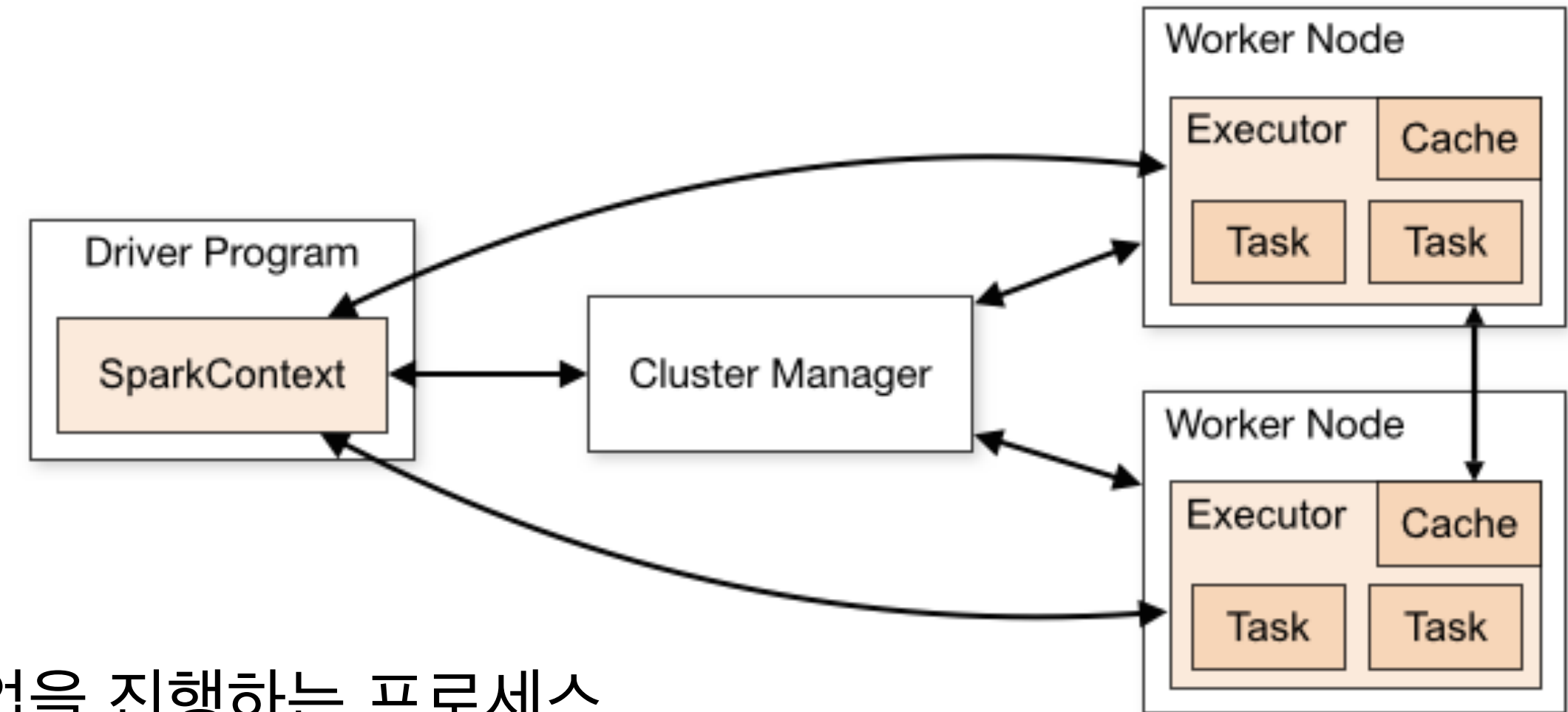


# Spark

## Spark 구조

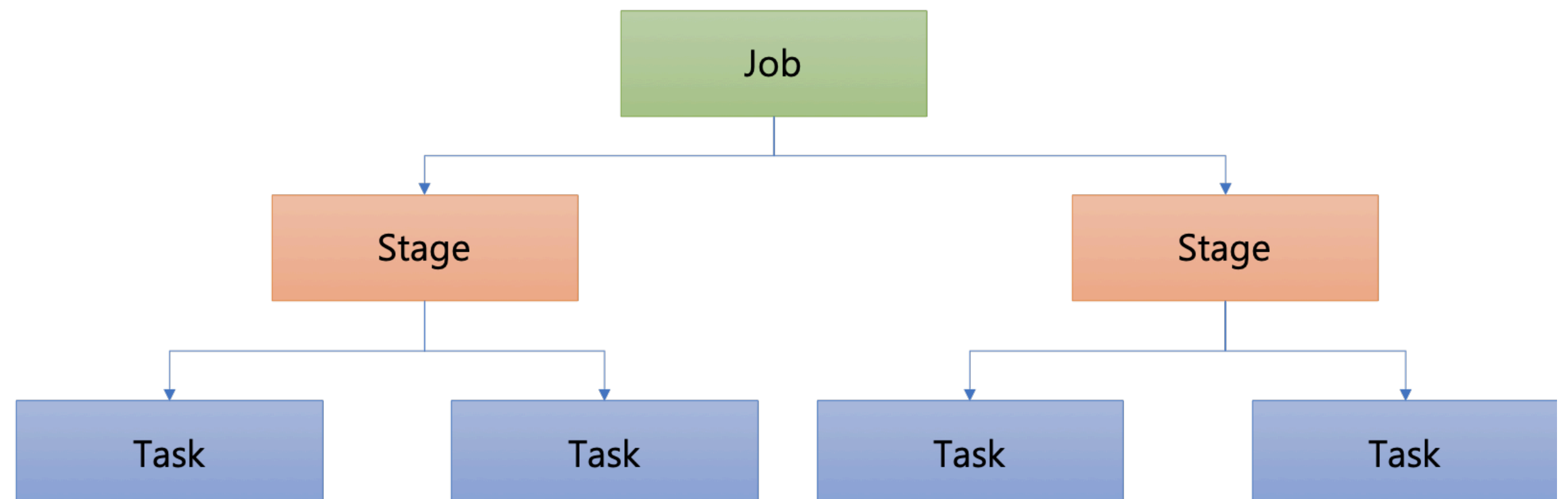
- Spark Application

- Driver: 스파크 애플리케이션을 실행하는 프로세스
- Executor: 테스크 실행을 담당하는 에이전트로 실제 작업을 진행하는 프로세스
- Task: Executor에서 실행되는 실제 작업



- Spark Job

- Job: 스파크 애플리케이션으로 제출된 작업
- Stage: 잡을 작업의 단위에 따라 구분한 것
- Task: Executor에서 실제로 실행되는 작업



<https://spark.apache.org/docs/latest/cluster-overview.html>

<https://wikidocs.net/26630>

# Spark

## Spark Implementation

- RDD (Resilient Distributed Data)

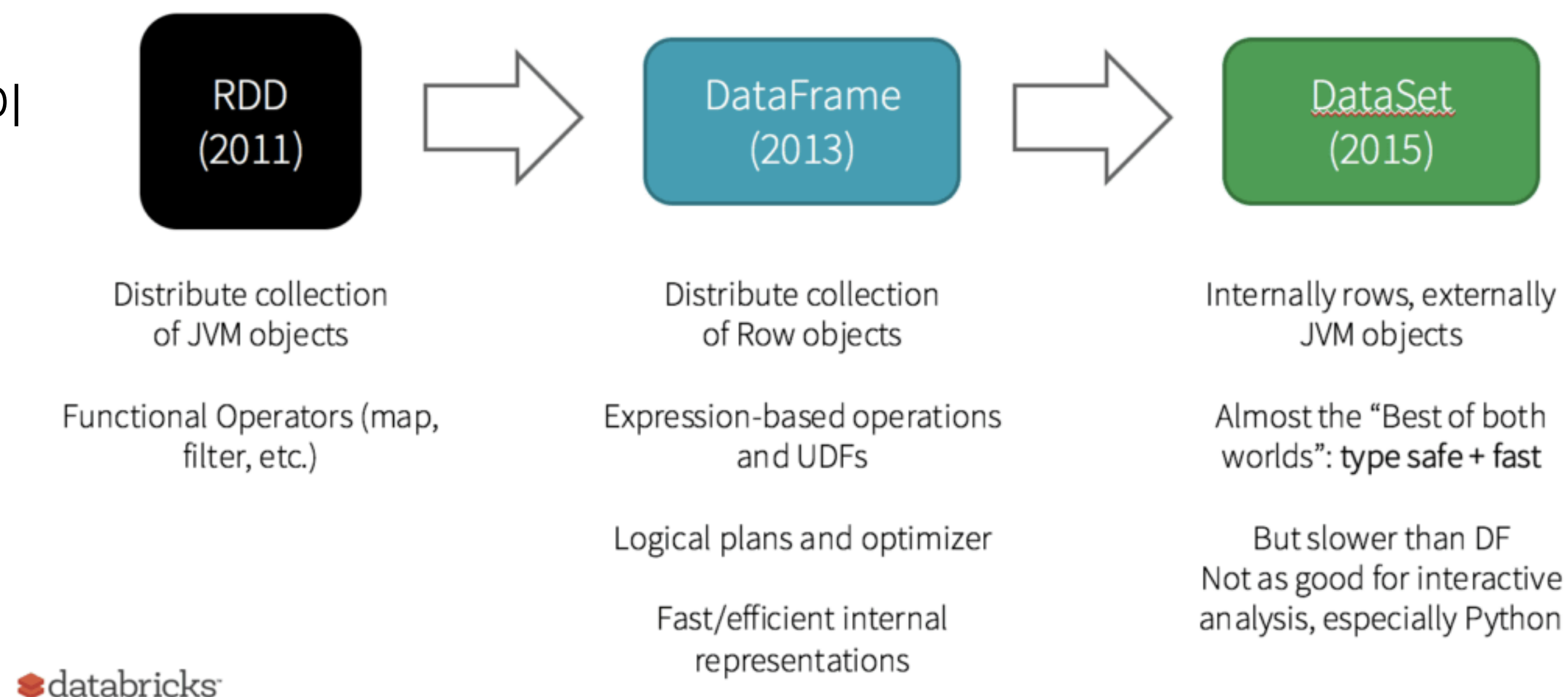
- 스파크 초기의 데이터 처리 구현 방식으로 인메모리 데이터 처리를 통하여 처리속도를 높였다는 점은 높이 살 수 있지만
- 사용자가 직접 처리를 해야하는 부분이 많이 남아 있었기 때문에 최적화에 어려움이 존재

- DataFrame

- RDD와 다르게 컬럼 기반으로 데이터가 정리되어 있어서, 사용자가 사용하기 훨씬 수월해짐

- Dataset

- 타입이 추가됨
- Spark 2.0에서 Dataframe과 통합



<https://databricks.com/glossary/what-is-rdd>

# References

- 빅데이터 - 하둡, 하이브로 시작하기, hs\_seo, <https://wikidocs.net/book/2203>
- 빅데이터 - 스칼라(scala), 스파크(spark)로 시작하기, hs\_seo, <https://wikidocs.net/book/2350>

**E.O.D**