

아파치 하둡 소개

순천향대학교 컴퓨터공학과
이 상 정

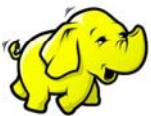
아파치 하둡소개

학습 내용

1. 아파치 하둡 소개
2. 하둡 에코 시스템

1. 아파치 하둡 소개

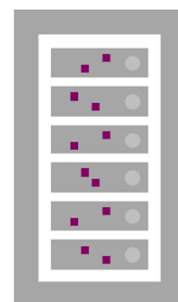
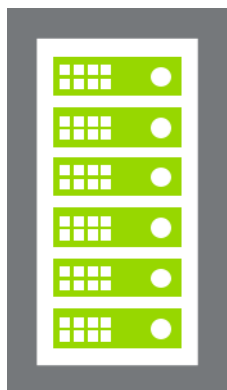
아파치 하둡소개



아파치 하둡 (Apache Hadoop) 소개

□ 아파치 하둡은 **대용량 데이터의 분산 저장 및 처리**를 위한 **오픈 소스 프레임워크**

- 특수한 전용 하드웨어가 아닌 **일반 범용 머신**들로 클러스터의 노드 구성
- 여분의 **리던던시(redundancy)**로 구현되어 높은 **고장 감내(fault-tolerant)** 시스템
 - 한 디스크가 노드가 고장 나도 데이터 손실 없이 나머지 노드가 작업을 수행



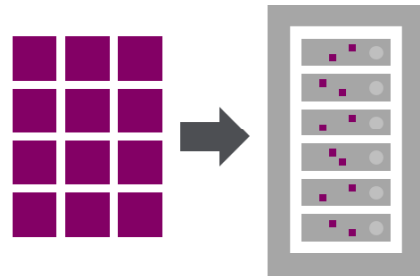
하둡 분산 파일 시스템 (HDFS)

하둡 분산 파일 시스템 (Hadoop Distributed File System)

- 디폴트로 데이터가 3번 복제(replication)
 - 최소 하나는 다른 노드에 위치



- 대용량 파일이 **블록**으로 분리
 - 블록 단위로 읽기/쓰기, 복제



컴퓨터 구조 복습 - 메시지 전달 멀티프로세서 덧셈 예

- 자신의 전용 메모리를 갖는 100개의 프로세서로 구성된 메시지 전달 멀티프로세서에서 100,000개의 숫자를 더하는 프로그램
 - 100,000개의 숫자를 모두 가지고 있는 프로세서가 100개의 프로세서-메모리 노드에 1000개의 숫자를 배분하여 전송
 - 각 프로세서들이 1000개의 합을 병렬로 더한 후, 각 부분합들을 프로세서에 전송하여 다시 더하는(reduction) 두 단계로 작성

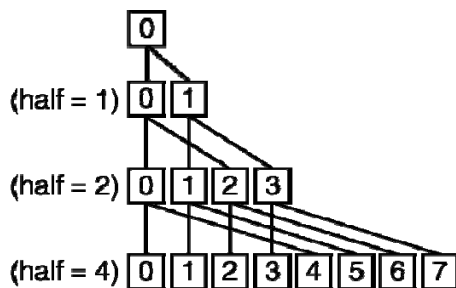
□ 답: 1000개의 숫자 병렬 덧셈 - 부분합

```
sum = 0;
for (i = 0; i < 1000; i = i + 1)
    sum = sum + AN[i];
```

컴퓨터 구조 복습 - Reduction (1)

□ 답: 리덕션(reduction)

- 분할정복(divide-and-conquer) 알고리즘을 사용
- 절반의 프로세서들은 부분합을 전송하고 나머지 절반은 수신하여 덧셈
- 다시 $\frac{1}{4}$ 의 프로세서들이 전송하고 나머지 $\frac{1}{4}$ 은 수신하여 덧셈
- 최종적으로 한 개의 결과가 나올 때까지 반복
- 전송과 수신 과정은 통신뿐만 아니라 동기화 수단으로도 사용



컴퓨터 구조 복습 - Reduction (2)

```
limit = 100; half = 100; /* 100 processors */
```

```
repeat
```

```
  half = (half+1)/2; /* send vs. receive
                    dividing line */
```

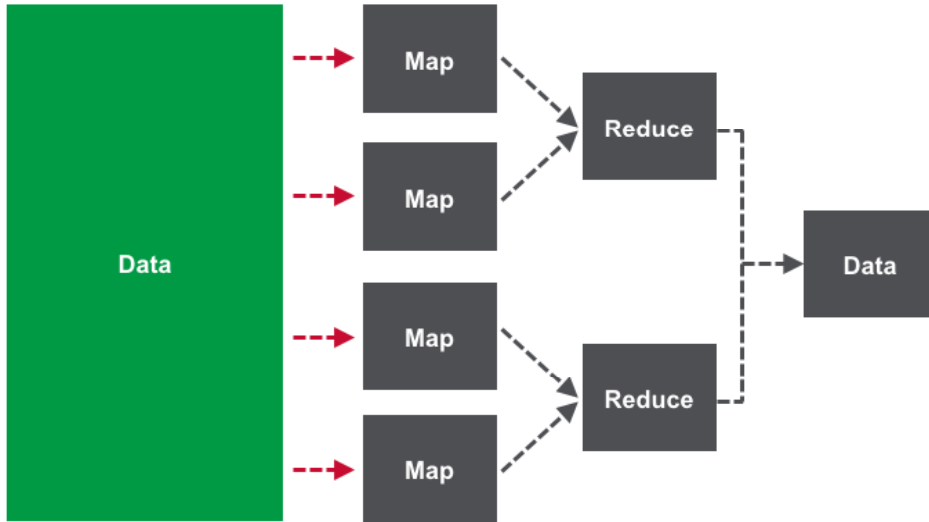
```
  if (Pn >= half && Pn < limit)
    send(Pn - half, sum);
```

```
  if (Pn < (limit/2))
    sum = sum + receive();
```

```
  limit = half; /* upper limit of senders */
until (half == 1); /* exit with final sum */
```

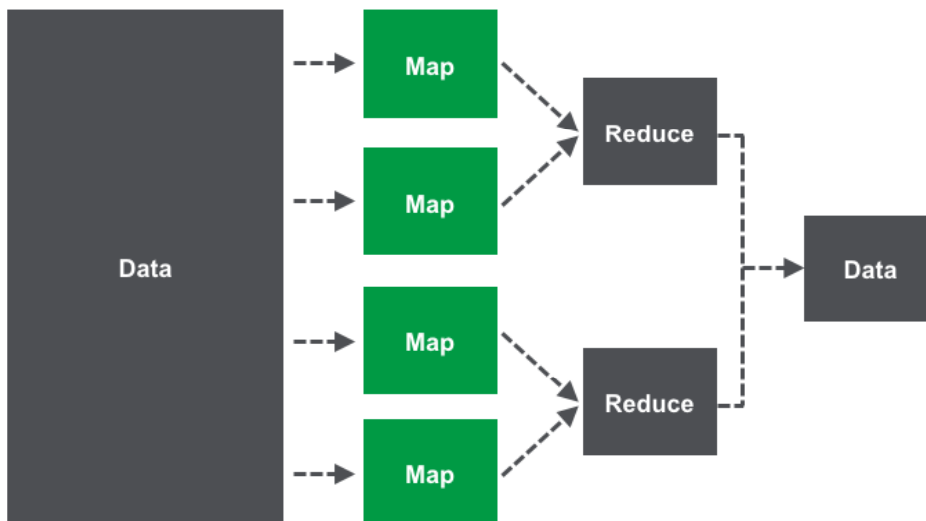
맵리듀스 (MapReduce) 알고리즘

- 맵리듀스 알고리즘은 맵(Map), 셔플(Shuffle), 리듀스(Reduce)의 3 단계로 구성



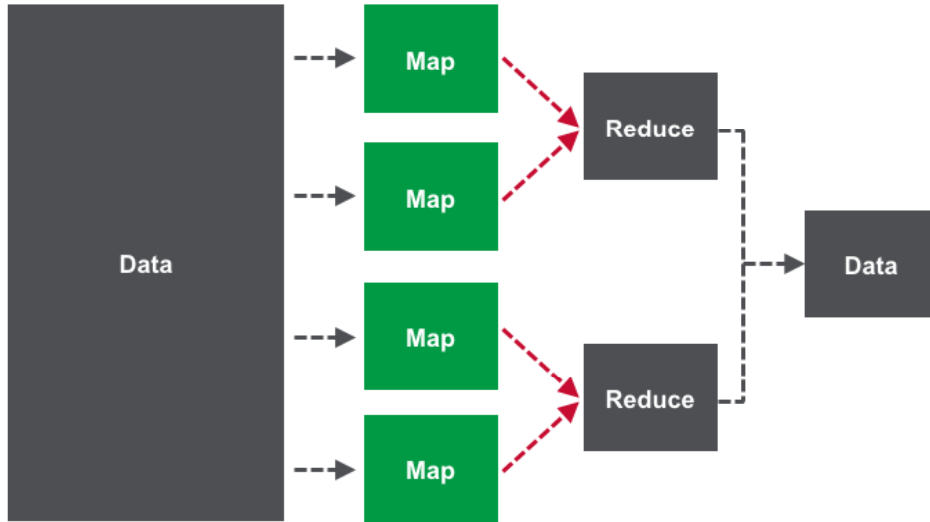
맵리듀스 알고리즘 - 맵 (Map)

- 맵 단계에서는 데이터를 분할하고, 맵 함수를 적용하여 키-값 쌍 (key-value pair)을 생성



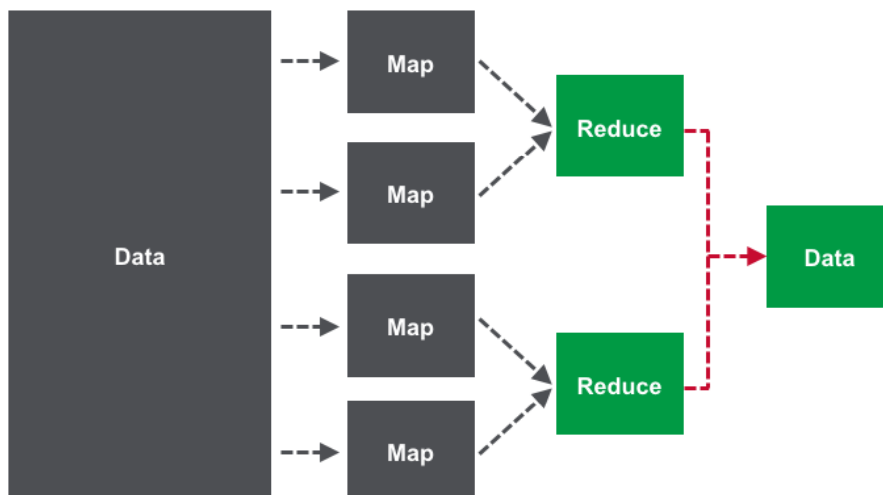
맵리듀스 알고리즘 - 셔플 (Shuffle)

- 셔플 단계에서는 맵 단계 출력의 키/값들의 쌍들을 정렬하고 분할하여 리듀스 단계로 전달



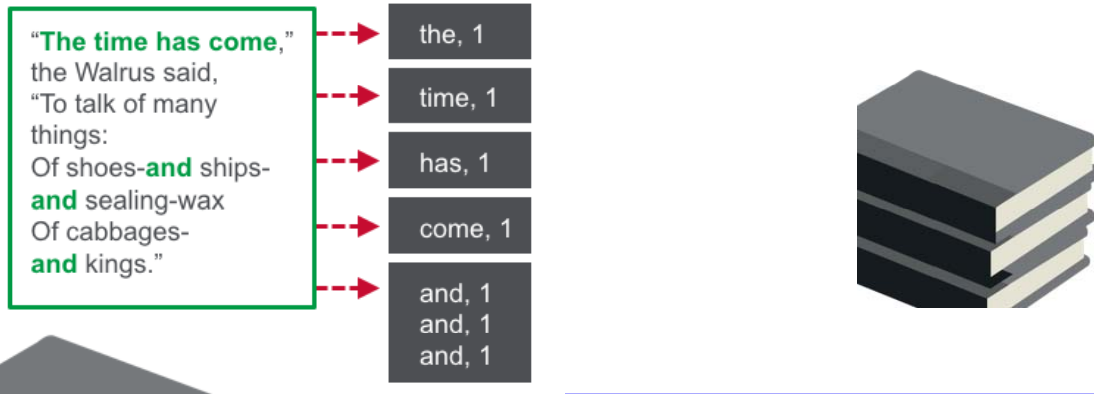
맵리듀스 알고리즘 - 리듀스 (Reduce)

- 리듀스 단계에서는 각 분할에 대해 리듀스 함수를 적용
- 맵리듀스 알고리즘은 분할정복 (divide-and-conquer) 기반 알고리즘으로 하나의 큰 작업을 여러개의 작은 작업으로 나누어 병렬 처리



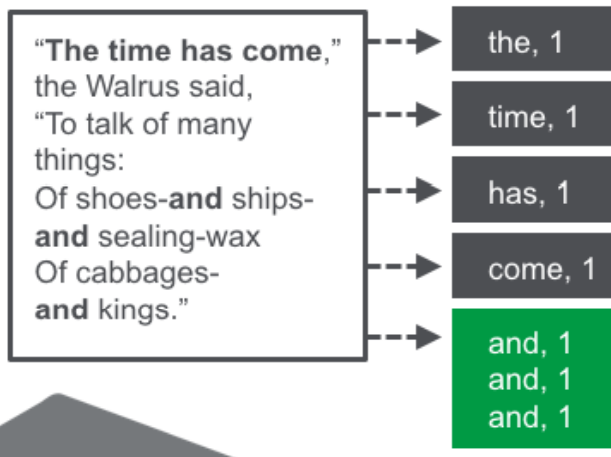
맵리듀스 단어 카운트 예 - 입력

- 맵리듀스 알고리즘의 “Hello World” 버전인 **단어 카운트 (word count)** 예로 동작 설명
 - Lewis Carroll 의 모든 책에 대한 단어를 카운트
- 하둡은 먼저 **데이터 파일들을 분할하여 데이터 노드들에 분배**
 - 분배 받은 한 노드 예



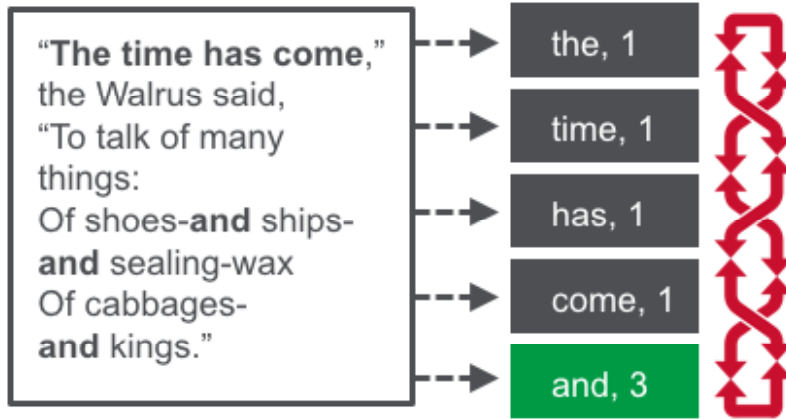
맵리듀스 단어 카운트 예 - 맵

- **맵 단계에서 입력 문자열을 토큰화하여 모든 단어에 대해 키-값 쌍을 출력**
 - 이 예에서는 3개의 “and”가 3개의 다른 키-값 쌍으로 생성



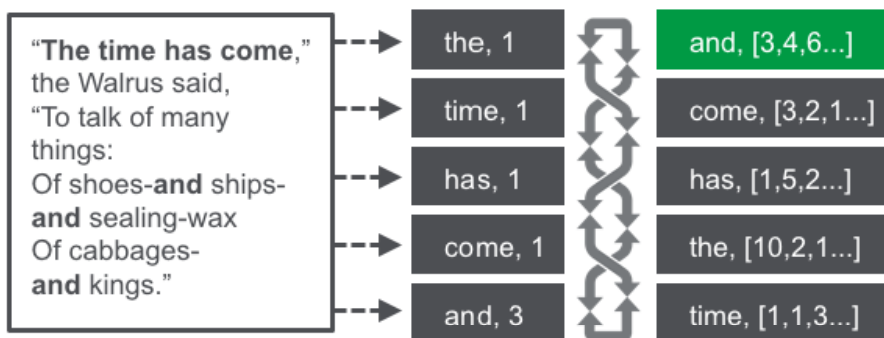
맵리듀스 단어 카운트 예 - 병합 및 셔플

- 병합 및 셔플 단계에서는 중복된 같은 키-값 쌍을 병합한 후에 셔플



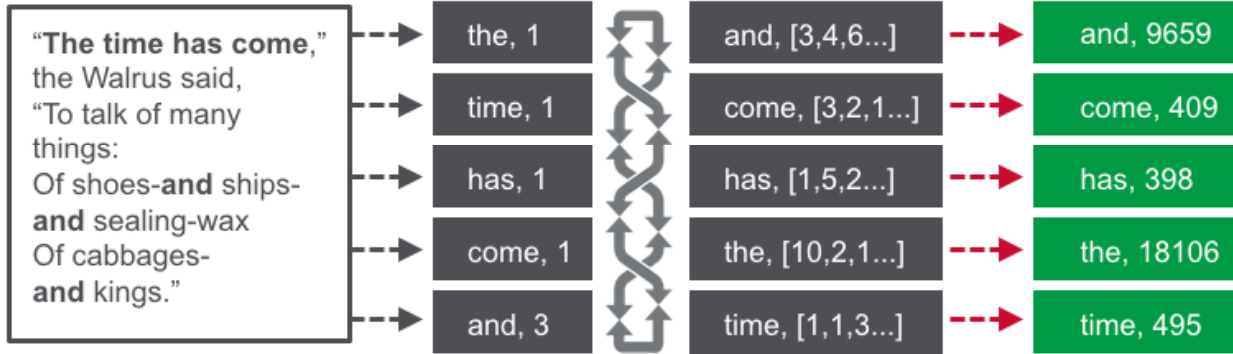
맵리듀스 단어 카운트 예 - 리듀스

- 리듀스 단계에서 다른 노드의 맵 결과들을 취합하여 통합



맵리듀스 단어 카운트 예 - 출력

- 마지막으로 하둡 프레임워크는 출력을 취합하여 파일 시스템에 저장



2. 하둡 에코 시스템

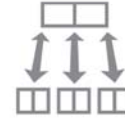
하둡 에코 시스템

하둡의 핵심 2 요소

- HDFS: 데이터 저장
- 맵리듀스: 데이터의 처리



HDFS



MapReduce



Unstructured



Semi-Structured



Structured

하둡은 다양한 소스의 구조화/비구조화 데이터를 하둡 클러스터에 가져와 처리하는 응용들을 제공

- 이들 데이터 처리 응용들의 일부는 하둡 에코 시스템 (Hadoop echo system)으로 알려짐



하둡 에코 시스템 - 빅 데이터 파이프라인

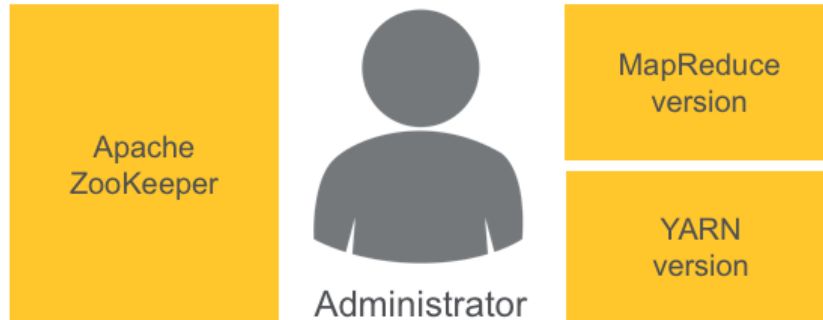
- 하둡 에코 시스템은 시스템 관리자, 프로그램 개발자, 데이터 분석가 등이 빅 데이터를 데이터의 소스로 부터 수집 (ingestion) 되어 처리 (processing) 및 분석 (analysis) 의 파이프라인 단계의 과정을 수행할 수 있는 툴을 제공



하둡 에코 시스템 - 클러스터 관리

- 하둡 관리자는 주키퍼(Zookeeper)를 사용하여 클러스터를 관리하고, 클러스터에 실행되고 있는 맵리듀스나 YARN의 버전을 파악하고 있어야 함

Responsible for many aspects of cluster maintenance



아파치 주키퍼 (Apache Zookeeper)

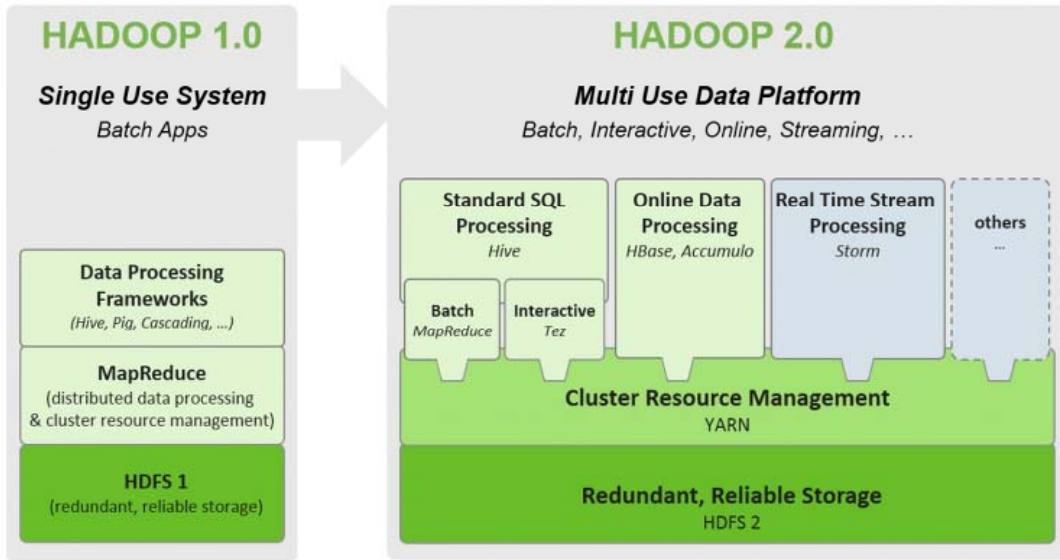
- 아파치 주키퍼는 클러스터의 각 노드들 사이의 서비스들을 관리하고 조정
 - 하둡에서 처음 실행되는 서비스
 - 서비스들 간의 동기화
 - 장애 상황 판단 및 복구
 - 네임서비스를 통한 부하분산
 - 환경 설정 관리



- Coordinates other services**
 - Maintains configuration information
 - Monitors cluster heartbeats
- Runs on an odd number of nodes**
 - Requires a majority for tie-breakers
- Starts before other services**

□ YARN (Yet Another Resource Negotiator)

- 하둡 클러스터의 각 응용에 **자원을 할당하고 모니터링**
- 다양한 응용들이 하둡 클러스터의 자원의 공유 지원



하둡 에코 시스템 - 작업 기술 및 데이터 수집

- 개발자는 자바, 파이썬, 스칼라 등을 사용하여 수행할 작업을 기술하고, 플럼(Flume), 우지(Oozie), 스콥(Sqoop) 등의 틀을 사용하여 데이터를 수집

Responsible for data ingestion



아파치 플럼 (Apache Flume)

- 아파치 플럼은 하둡 클러스터에 **스트리밍 데이터를 수집**하는 오픈소스 서비스
 - 시스템 로그(syslog), 웹 서버 로그, 소셜 미디어 피드 등의 외부 소스로 부터의 스트림 이벤트 데이터들을 하둡 클러스터에 가져옴



- Ingests data into Hadoop cluster
- Commonly used with log files

아파치 스쿱 (Apache Sqoop)

- 아파치 스쿱은 하둡 클러스터와 외부 데이터 저장소 (RDBMS, NoSQL 데이터) 사이에 데이터를 가져오고 내보내는 툴



- Transfers data between HDFS and external data stores
- Can be used with RDBMS and NoSQL

아파치 우지 (Apache Oozie)

- 아파치 우지는 하둡의 워크 플로우(workflow, 작업 흐름)를 생성하고 스케줄링하는 툴
 - 워크플로우에는 자바나 스크립트 뿐만 아니라 맬리듀스, 피그(Pig), 하이브(Hive), 스쿱(Sqoop) 등과 같은 다수의 수행 작업들이 포함



- Schedules workflows
- Manages multiple jobs

하둡 에코 시스템 - 데이터 처리

- 개발자는 데이터 수집 뿐만 아니라 스파크(Spark), HBase, 피그(Pig) 등의 툴을 사용하여 데이터를 처리

Responsible for data processing



Developer

Spark

HBase

Pig

아파치 스파크 (Apache Spark)

- 아파치 스파크는 하둡 등과 같은 분산 컴퓨팅 클러스터 상에서 범용 데이터 분석을 수행하는 프레임워크
 - 메모리 상에 캐시된 데이터를 사용하여 반복적인 작업을 수행하여 기계학습 등과 같은 복잡한 분석을 매퍼듀스 보다 빨리 처리
 - 스칼라(Scala), 파이썬, 자바로 응용을 작성



- Iterative, in-memory jobs
 - Cached and fault-tolerant
- Written in Java, Scala, or Python

아파치 HBase

- 아파치 HBase는 구글의 빅테이블 모델을 따라 개발된 오픈 소스 NoSQL 데이터베이스
 - 시스템 측정 값, 사용자 클릭 등과 같은 대용량 데이터를 행과 열들을 저장
 - 채팅, 이메일과 같은 웹 일관성이 없는 값들을 갖는 데이터를 열에 저장
 - 웹 응용이나 검색 색인과 같이 랜덤 읽기/쓰기 접근이 연속으로 발생하는 데이터 저장



- NoSQL database
 - Capture large amounts of data
 - Store sparse data with inconsistent values
 - Continuous data with read/write access
- Written in Java

아파치 피그 (Apache Pig)

□ 아파치 피그는 데이터를 분석하는 플랫폼

- 데이터 플로우 스크립트 언어 “Pig Latin” 제공
- 스크립트를 맵리듀스 프로그램 시퀀스로 변환
- 파이썬, 자바, 자바스크립트, 루비로 함수 작성



- Includes scripting language called Pig Latin
- Transform and analyze large datasets
- Write functions in Python, Java, JavaScript, Ruby

하둡 에코 시스템 - 데이터 분석

□ 데이터 분석가는 하이브(Hive), 드릴(Drill), 머하웃(Mahout) 등의 툴을 사용하여 데이터를 분석

Responsible for analyzing data



Hive

Drill

Mahout

Data Analyst

아파치 하이브 (Apache Hive)

□ 아파치 하이브는 하둡 상에서 구축되는 데이터웨어하우스 (data warehouse) 인프라스트럭처

- HCatalog를 사용하여 데이터를 테이블에 저장하고, Hive Metastore를 사용하여 데이터를 추적
- HiveQL이라는 SQL과 같은 질의 언어를 사용하여 맵리듀스 프로그램을 생성하고 실행



- Stores data in HCatalog and Hive Metastore
- Hive Query Language (HQL)
 - SQL-like query language
- Uses MapReduce

아파치 드릴 (Apache Drill)

□ 아파치 드릴은 빅 데이터 탐색을 위한 질의 엔진

- SQL을 사용하여 HDFS나 하이브에 저장된 구조화, 반구조화, 비구조화 데이터들에 대한 동적 질의를 수행



- Structured, semi-structured, and unstructured data
- Multiple data sources and data types
- ANSI-SQL compliant

아파치 머하웃 (Apache Mahout)

□ 아파치 머하웃은 기계학습 라이브러리

- 군집화(clustering), 분류(classification), 협업 필터링(collaborative filtering) 알고리즘 지원
- 대규모 빅데이터를 알고리즘에 적용하여 추천 등과 같은 예측을 시도



- Clustering
- Classification
- Collaborative filtering

과제

□ 아파치 하둡 에코 시스템 중 하나를 선택하여 조사

- 앞에서 소개한 또는 소개하지 않은 에코 시스템
- 맵리듀스, YARN 포함

- MapR Academy, <http://learn.mapr.com/>
 - Introduction to Big Data
 - <http://learn.mapr.com/ess-100-introduction-to-big-data>