LESSON 1

스파크(Apache Spark) 개념

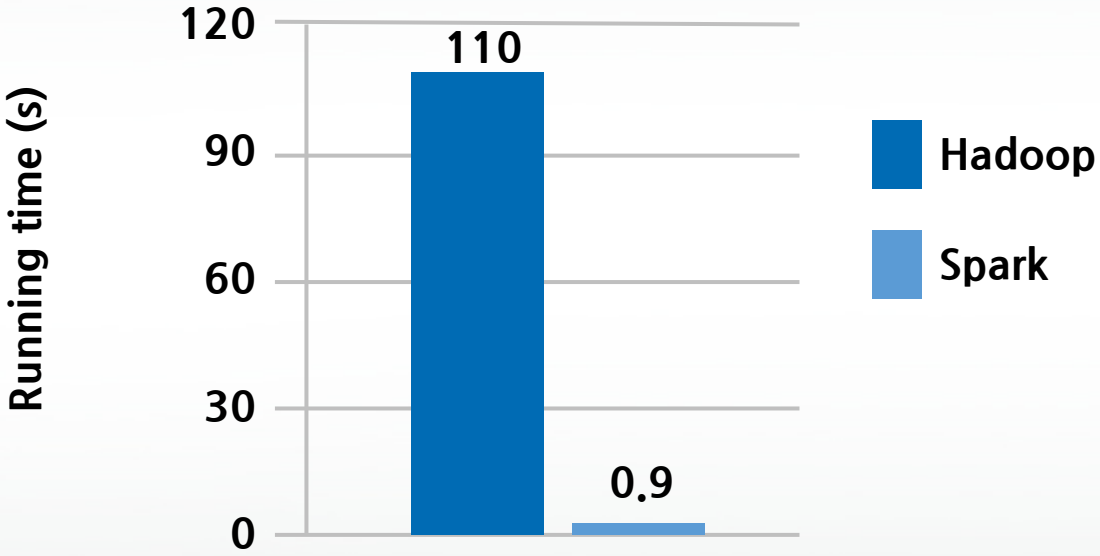
스파크(Apache Spark) 개념

스파크(Apache Spark)란?

- UC 버클리의 AMPlab에서 만든 경량 오픈소스 분산처리 프레임워크
  - Spark: Cluster Computing with Working Sets
- 단순 맵리듀스 외에 SQL/스트리밍/머신러닝이 묶인 구조
- 메모리를 최대로 활용해 반복작업에 높은 효율
  - 하둡은 디스크기반
- 스칼라(Scala) 언어로 되어 있음
  - 자바, 파이썬 지원
  - Spark SQL에서 Language-Integrated queries는 스칼라만,
  - Spark Streaming은 스칼라와 자바,
  - MLlib의 각종 Matrix는 스칼라와 자바에서 지원한다.
  - 셸은 스칼라와 파이썬만 지원한다.

스파크(Apache Spark) 개념

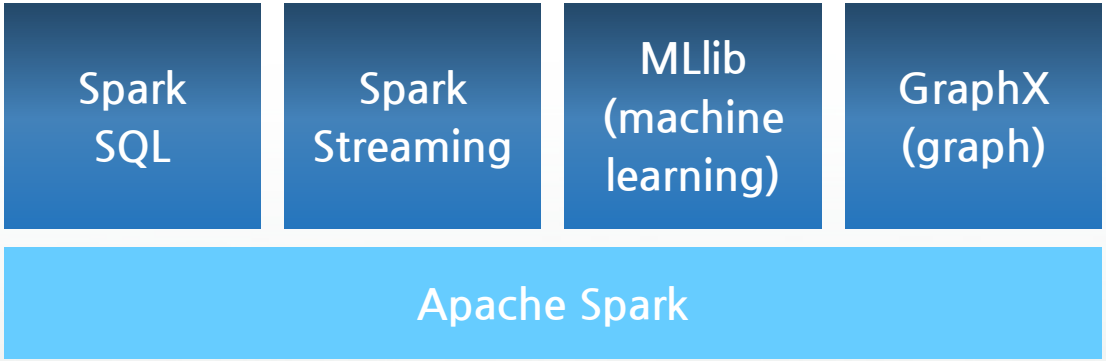
스파크(Apache Spark)란?



스파크(Apache Spark) 개념

스파크아키텍처

- Low latency (interactive) queries on historical data
- Low latency queries on live data(streaming)
- Sophisticated data processing



스파크(Apache Spark) 개념

RDD

- ‘Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing’
  - UC Berkeley, 2012년
- Read Only
  - 데이터를 수정 가능하면 데이터 유실 시 복구가 어려워짐
- Fault Tolerant
  - 데이터 유실 시 복구가능
  - Mapped RDD 중에 일부 데이터가 손실되면 데이터 도출 경로(Lineage)를 통해 재계산을 해서 해당 데이터를 다시 복구
- Lazy Loading
  - 실제 사용할 때 로딩

스파크(Apache Spark) 개념

RDD

- Resilient Distributed Dataset
- 메모리(RAM)에 읽기 전용, 파티션되어 있는 자료
  - Immutable, partitioned collections of records
- 부모로부터 생성된 이력(lineage)만 기록
- Fault Tolerant 특성 지원
- Transformation
  - 기존의 RDD에서 새로운 RDD생성(필터링, 맵, 리듀스 등)
  - 실제 계산이 되지는 않음
  - Lineage에 DAG그림
- Action
  - RDD기반으로 실제 계산이 되도록 지정(count와 같은 함수)

스파크(Apache Spark) 개념

RDD

- Lazy Loading
  - 실제 데이터는 마지막 단계에서 로딩
  - load명령이 아닌 실제 action명령이 수행되면 그 때 로딩

스파크(Apache Spark) 개념

스파크 아키텍처

드라이버 프로그램(Driver Program)

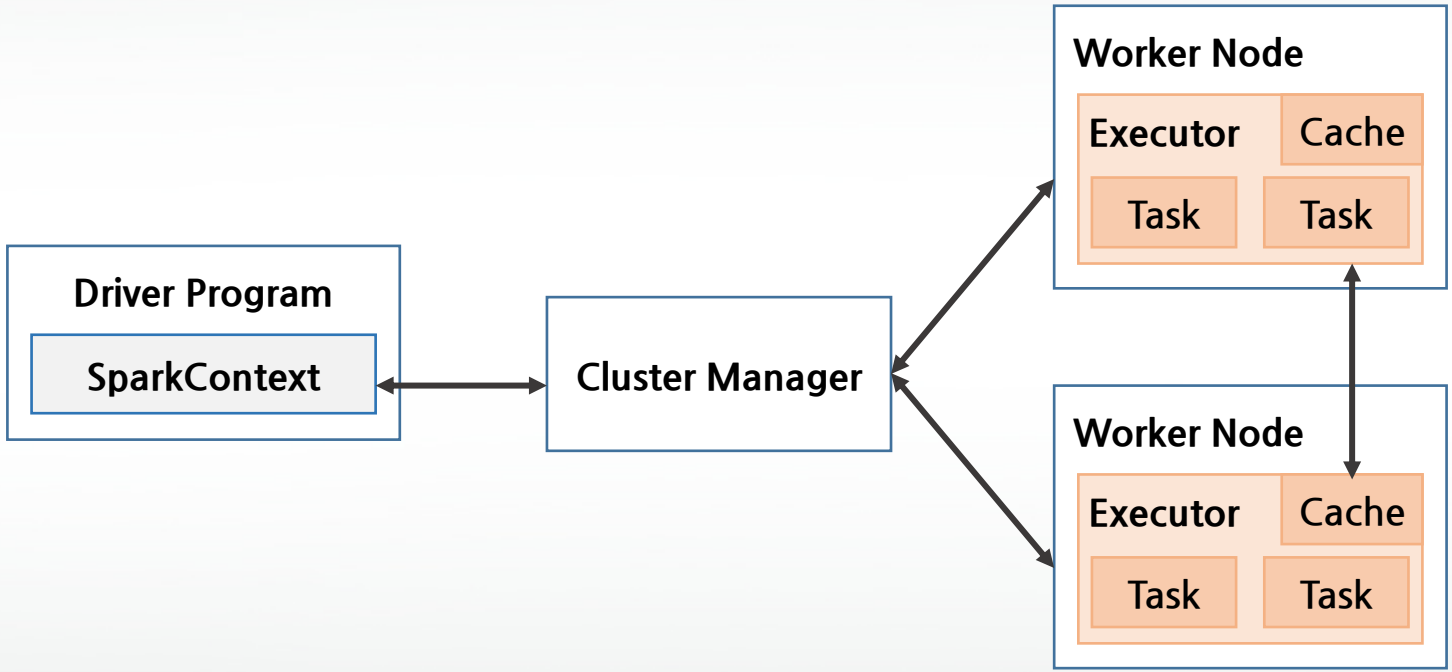
- SparkContext가 포함된 메인 프로그램
- RDDs의 transform이나 조작을 정의해 놓고, 스파크 마스터에 요청해서 클러스터 매니저를 통해 워커 노드에서 실제적인 처리가 되도록 함

워커노드(Worker Node)

- 실제로 데이터를 처리하는 노드
- 프로그램의 요청이 오면 각각 Executor를 실행
- 내부에서 태스크와 캐시를 생성
- 프로그램의 코드를 받아 실제적인 데이터처리 한 후 결과값을 저장

스파크(Apache Spark) 개념

스파크 아키텍처



LESSON 2

설치

설치

스칼라(Scala) 설치

- 스칼라설치
  - sudo apt-get install scala
- 환경변수 설정(~/.bashrc or ~/.profile)
  - export SCALA_HOME=/usr/bin

설치

Apache Spark 설치 & 세팅

다운로드

- wget http://d3kbcqa49mib13.cloudfront.net/spark-2.0.0-bin-hadoop2.7.tgz

압축해제

- tar xvfz spark-2.0.0-bin-hadoop2.7.tgz

설치

Apache Spark 세팅

- 환경설정(~/.bashrc or ~/.profile)
  - export SPARK_HOME = /spark-2.0.0-hadoop2.7
  - export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin

- conf/spark-env.sh
  - export SCALA_HOME = /usr/bin
  - export SPARK_HOME = /spark-2.0.0-hadoop2.7

설치

Apache Spark 세팅

스파크 서버실행

- master 실행/종료 start-master.sh/stop-master.sh
- slave 실행/종료 start-slaves.sh/stop-slaves.sh
- start-all.sh/stop-all.sh

HDFS실행

- start-dfs.sh

Jps

- 307 Jps
- 19 SparkSubmit
- 134 Master
- 252 Worker

쉘실행

- spark-shell(scala) / pyspark(python)

설치

Apache Spark예제(Scala)

HDFS 파일복사

- hadoop fs -mkdir /user
- hadoop fs -mkdir /user/root
- hadoop fs -copyFromLocal README.md /user/root/
- hadoo fs -ls /user/root

설치

Apache Spark예제(Scala)

```
val textFile = sc.textFile("README.md")

textFile.count()
textFile.first()

var lines = sc.textFile("README.md")
var lineLengths = lines.map(s => s.length)
var totalLength = lineLengths.reduce((a,b) => a+b)
```

설치

Apache Spark예제(Scala)

```
val file = sc.textFile(“readme.md”)
file.flatMap(line => line.split(" "))
    .map(word => (word,1))
    .reduceByKey(_ + _)
```