자바쌤~!

https://cafe.naver.com/javassem

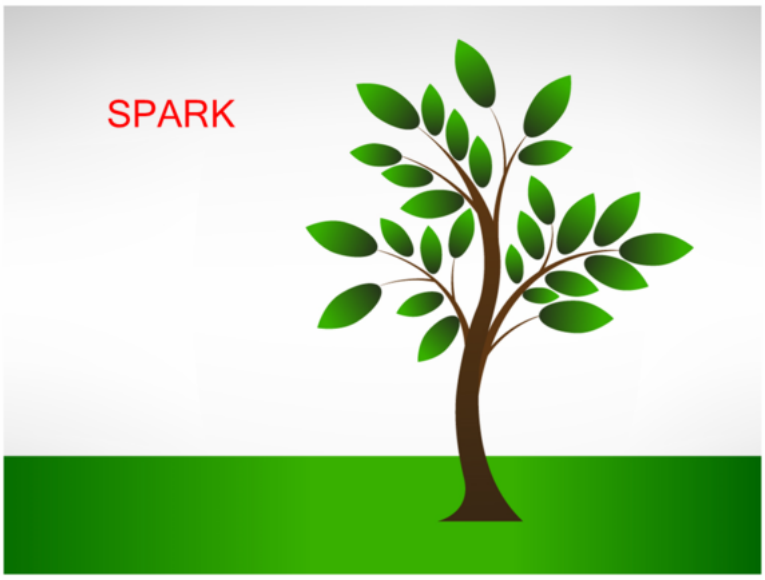
검색

63기-자바 & 빅데이타 >

[스팍] 개념

자바쌤 1:1 채팅
2020.06.25. 08:24 조회 35

댓글 0 URL 복사



카페정보 나의활동

매니저 자바쌤
since 2013.03.15.
카페소개

새싹2단계
436 초대하기

즐겨찾는 멤버 64명
게시판 구독수 4회
우리카페앱 수 0회

카페 글쓰기

카페 채팅

즐겨찾는 게시판

전체글보기 4,082

KOSMO

63기-자바 & 빅데이타

팁
- 심심문제
- 기타등등
- JAVA
- Oracle/Mysql
- JSP
- WebUI
- 파이썬
- Spring
- 리눅스
- 하둡+eco
- 프로젝트

공유해요
- 알고리즘도전
- 기술면접 준비하기
- 실존 면접
- 기회는 준비된자에게
- 프로젝트 팁

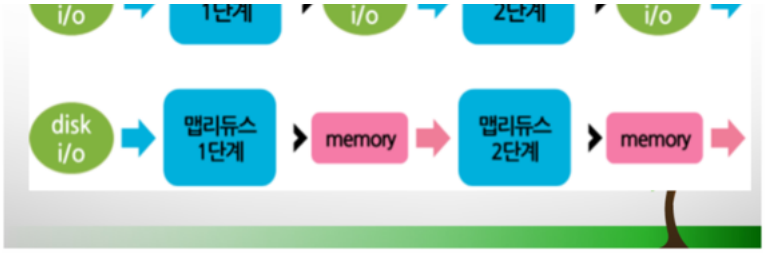
쉬어가기

청년아카데미

전문가과정

공지사항
카페회칙

카페탈퇴하기



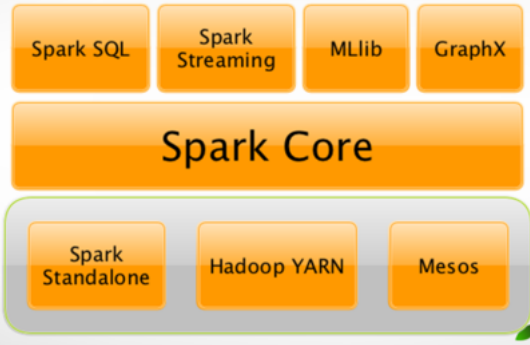
1. Spark 출현 배경

3. Spark 개발

- 기존 디스크 입출력에 대한 지연 시간 개선
- 메모리를 사용하여 반복적인 작업이나 스트리밍 데이터를 효율적으로 처리
- 그러나 메모리 사용으로 비용 증가
- 메모리의 특성인 비휘발성으로 중간 데이터 손실

2. 스파크 개요

1. 스파크의 주요 요소



2. 스파크 개요

1. 스파크의 주요 요소

- Spark SQL : sql 기반으로 쿼리 수행
- Spark Streaming : 데이터 스트림을 개별 세그먼트로 나눈 후 각 세그먼트의 데이터를 스파크 엔진으로 처리함
- Spark MLlib : 머신러닝 라이브러리 포함
- Spark GraphX : 그래프 라이브러리
- Spark Core : 스파크 전체의 기초가 되는 분산 작업 처리, 스케쥴링, API 인터페이스 (java, python, scala, R) 지원
- Spark Standalone : 스파크 단독 실행
- Yarn : 하둡과 연결하여 실행
- Mesos : 자체 개발한 자원관리

2. 스파크 개요

2. RDD 란?

- Resilient Distributed Dataset : 탄력적으로 분산된 데이타셋
- 스파크에서 사용되는 기본 데이터 구조
- RDD 특징

  - 다수의 파티션으로 분산 노드에 나눠져서 관리됨 (partitioned)

  - 변경이 불가능 (Immutable)

partitioned : 데이터 셋을 잘게 잘라서 분산
가장 효율적으로 클러스터 노드에 분산시킴
immutable : 만들어진 뒤에는 변하지 않기 때문에 만들어진 흐름만
알면 다시 만들 수 있음.

2. 스파크 개요

3. RDD를 제어하는 연산

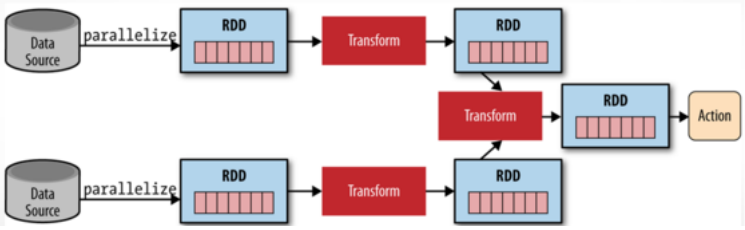
Transformation (변환)	RDD에서 새로운 RDD를 생성하는 함수
	- filter : 특정 데이터만 추출
	- map : 데이터를 분산 배치
Action (액션)	RDD에서 다른 타입의 데이터로 변환하는 함수
	- count : 변환 연산 후 파티션의 데이터 요소 개수
	- collect : 변환 연산 후 파티션의 데이터 요소 집합

[확인] http://spark.apache.org/docs/latest/rdd-programming-guide.html

2. 스파크 개요

Transformations	*map*(*f* : T ⇒ U)	:	RDD[T] ⇒ RDD[U]
	filter(*f* : T ⇒ Bool)	:	RDD[T] ⇒ RDD[T]
	flatMap(*f* : T ⇒ Seq[U])	:	RDD[T] ⇒ RDD[U]
	sample(*fraction* : Float)	:	RDD[T] ⇒ RDD[T] (Deterministic sampling)
	groupByKey()	:	RDD[(K, V)] ⇒ RDD[(K, Seq[V])]
	reduceByKey(*f* : (V, V) ⇒ V)	:	RDD[(K, V)] ⇒ RDD[(K, V)]
	union()	:	(RDD[T], RDD[T]) ⇒ RDD[T]
	join()	:	(RDD[(K, V)], RDD[(K, W)]) ⇒ RDD[(K, (V, W))]
	cogroup()	:	(RDD[(K, V)], RDD[(K, W)]) ⇒ RDD[(K, (Seq[V], Seq[W]))]
	crossProduct()	:	(RDD[T], RDD[U]) ⇒ RDD[(T, U)]
	mapValues(*f* : V ⇒ W)	:	RDD[(K, V)] ⇒ RDD[(K, W)] (Preserves partitioning)
	sort(*c* : Comparator[K])	:	RDD[(K, V)] ⇒ RDD[(K, V)]
	partitionBy(*p* : Partitioner[K])	:	RDD[(K, V)] ⇒ RDD[(K, V)]
Actions	*count*()	:	RDD[T] ⇒ Long
	collect()	:	RDD[T] ⇒ Seq[T]
	reduce(*f* : (T, T) ⇒ T)	:	RDD[T] ⇒ T
	lookup(*k* : K)	:	RDD[(K, V)] ⇒ Seq[V] (On hash/range partitioned RDDs)
	save(*path* : String)	:	Outputs RDD to a storage system, *e.g.*, HDFS

2. 스파크 개요



2. 스파크 개요

4. 스파크 특징

인 메모리 기반의 대용량 데이터 처리

API 인터페이스 (java, python, scala, R) 지원

스파크 단일노드, 하둡 YARN 및 mesos 등의 다양한 환경에서 작동

5. 스파크 활용

전자상거래 데이터 수집을 통한 고객 매출 최적화

사용자와 상호작용할 수 있는 예측 모델 구성

3. 스파크 실행

0. 스파크 실행

Spark Start 을 실행하는 명령어 : start-all.sh
Spark Start 을 종료하는 명령어 : stop-all.sh

[*주의*]
hadoop start와 이름이 같으므로 반드시 해당 경로로 이동후 실행하기

[hadoop@dn01 ~]$ cd /opt/spart/current/sbin

[hadoop@dn01 sbin]$ ./start-all.sh

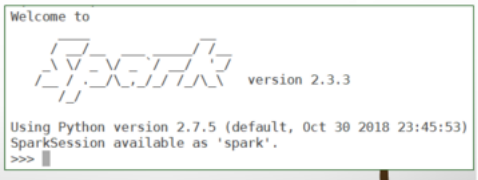
[*주의*] 현재디렉토리표시 점(.)으로 지정하고 명령어 호출

3. 스파크 실행

1. 스파크 대화용 프로그램

(1) scala : spark-shell

(2) pyspark : pyspark



3. 스파크 실행

2. 스파크 응용 프로그램

- 파이썬 application실행

[hadoop@dn01 ~]$ spark-submit --master local[4] test.py

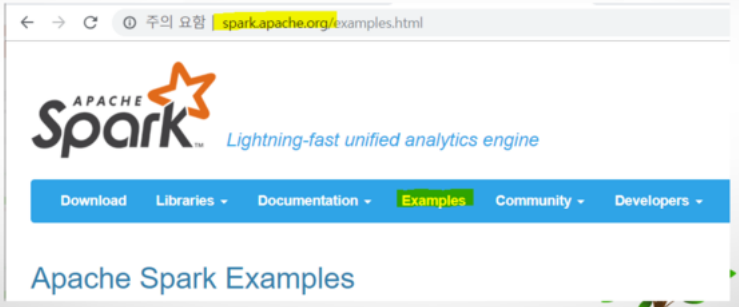
- 스칼라 : sbt로 빌드 (. jar)
- 자바 : eclipse Maven 으로 빌드 (.jar)

[hadoop@dn01 ~]$ spark-submit -class "TestApp" --master local[4] test-project.jar

3. 스파크 실행

3. 샘플 확인

http://spark.apache.org > Examples



3. 스파크 실행

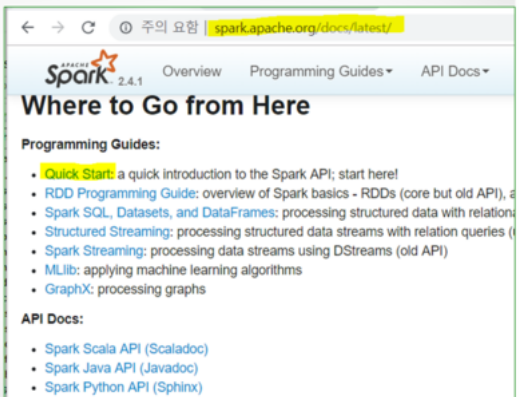
3. 샘플 확인

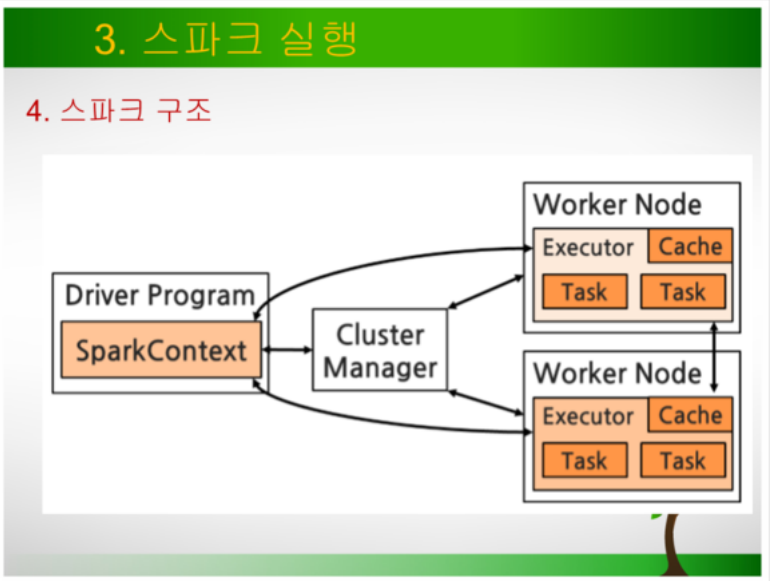
http://spark.apache.org > Examples

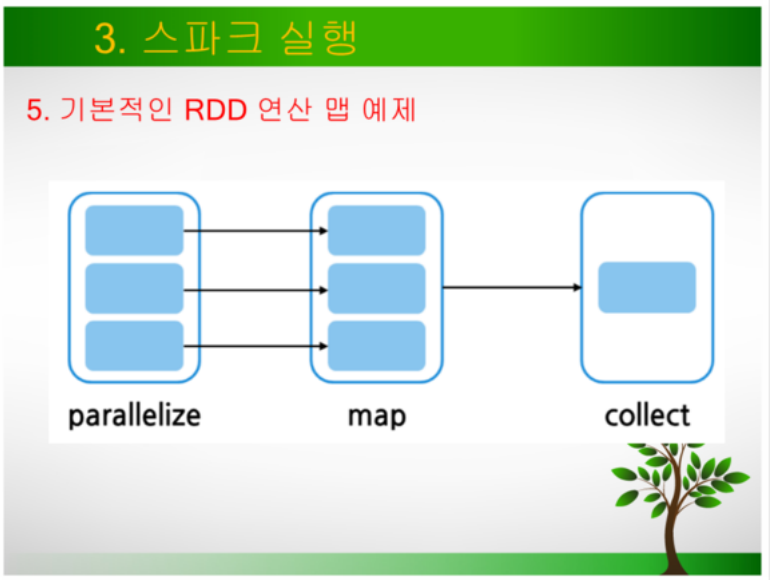
3. 스파크 실행

3. 샘플 확인

http://spark.apache.org > Documetation







댓글 댓글알림

김용선

댓글을 남겨보세요

등록