스파크 RDD

1 RDD 개념

① RDD(Resilient Distributed Dataset) 신규 개념 도입

스파크에서 사용되는 기본 데이터 구조

스파크에서 내부적으로 연산하는 데이터들을 모두 RDD 타입으로 처리

Immutable, Partitioned Collections of Record

1. 여러 분산 노드에 나누어짐
2. 다수의 파티션으로 관리됨
3. 변경이 불가능한 데이터 셋

스파크 RDD

2 RDD 생성

1 RDD의 생성

외부로부터 데이터를 로딩할 때

코드에서 생성되는 데이터를 저장할 때

3 RDD를 제어하는 2개의 연산 타입

1 RDD를 제어하는 2개의 연산 타입

Transformation(변환)

✓ RDD에서 새로운 RDD를 생성하는 함수

- filter : 특정 데이터만 산출하는 연산자
- map : 데이터를 분산 배치하는 연산자

Action(액션)

✓ RDD에서 RDD가 아닌 다른 타입의 데이터로 변환하는 함수들

- count() : 변환 연산 후 파티션의 데이터 요소 개수
- collect() : 변환 연산 후 파티션의 데이터 요소 집합

스파크 RDD

4 RDD 분산 처리

① RDD 분산 처리 방법

Immutable : 만들어진 뒤에는 변하지 않음

어떻게 만들었는지 알면 또 만들 수 있음

Partitioned : 데이터 셋을 잘게 잘라서 분산

가장 효율적으로 클러스터 노드에 분산시켜 볼 수 있음

② 파티션(Partition)

하나의 RDD는 여러 개의 파티션으로 나뉘어짐

- 성능에 유효한 영향을 줌

파티션의 개수, 기본 파티셔너(Hash, Range) 선택 가능

- 기본 파티셔너 외에도 사용자가 정의한 파티셔너 사용가능

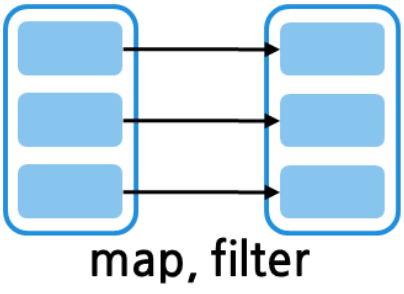
스파크 RDD

4 RDD 분산 처리

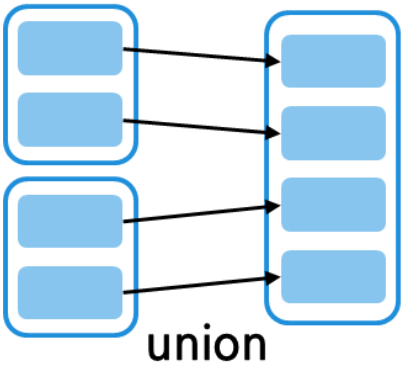
3 Dependency 타입

Narrow Dependency

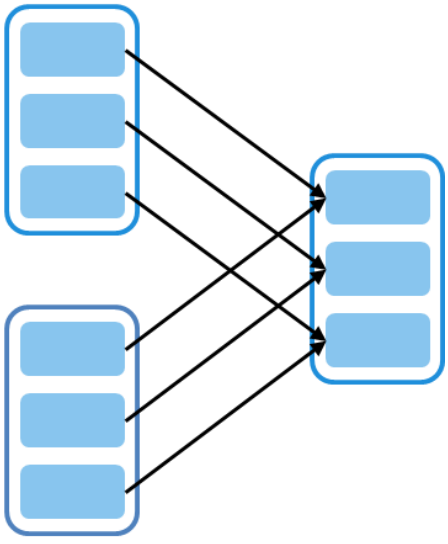
- 변환 후 파티션이 하나의 파티션으로 매핑되는 경우
- 해당 작업이 한 노드에서 처리 가능
- 네트워크를 사용하지 않음
- 파티션에 문제가 생겨도 복원 가능



map, filter



union

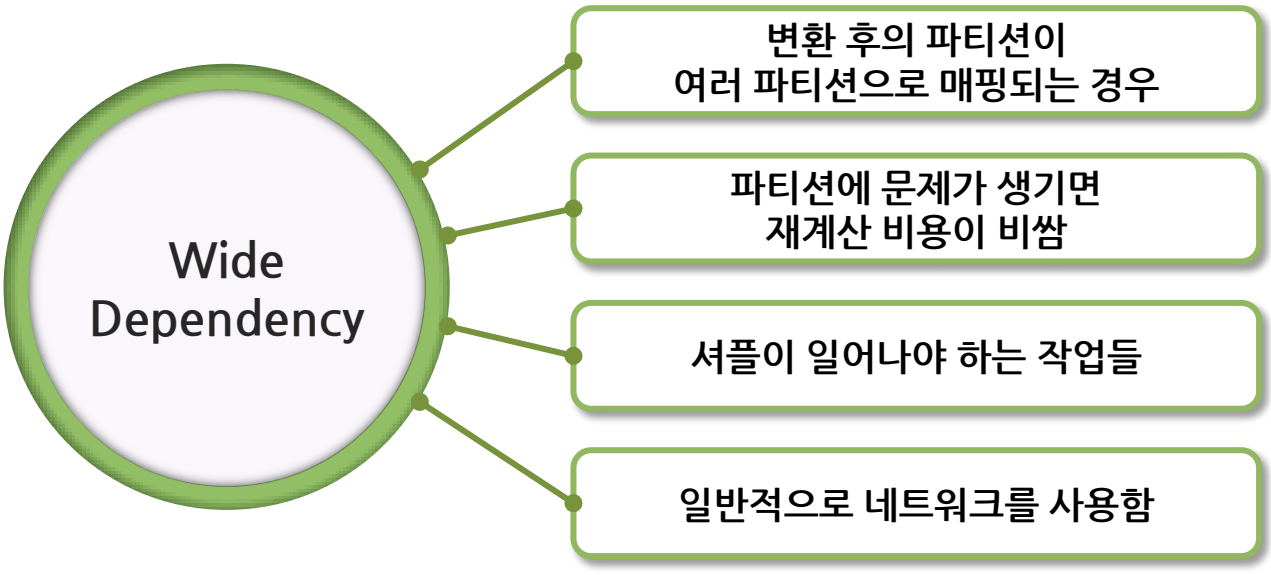


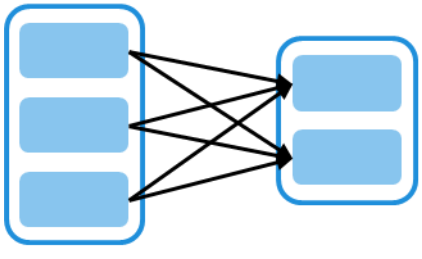
Join with inputs co-partitioned

스파크 RDD

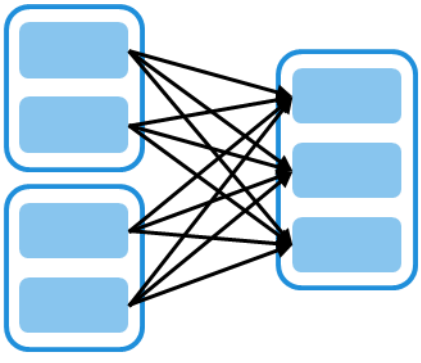
4 RDD 분산 처리

3 Dependency 타입





groupByKey



Join with inputs not
co-partitioned

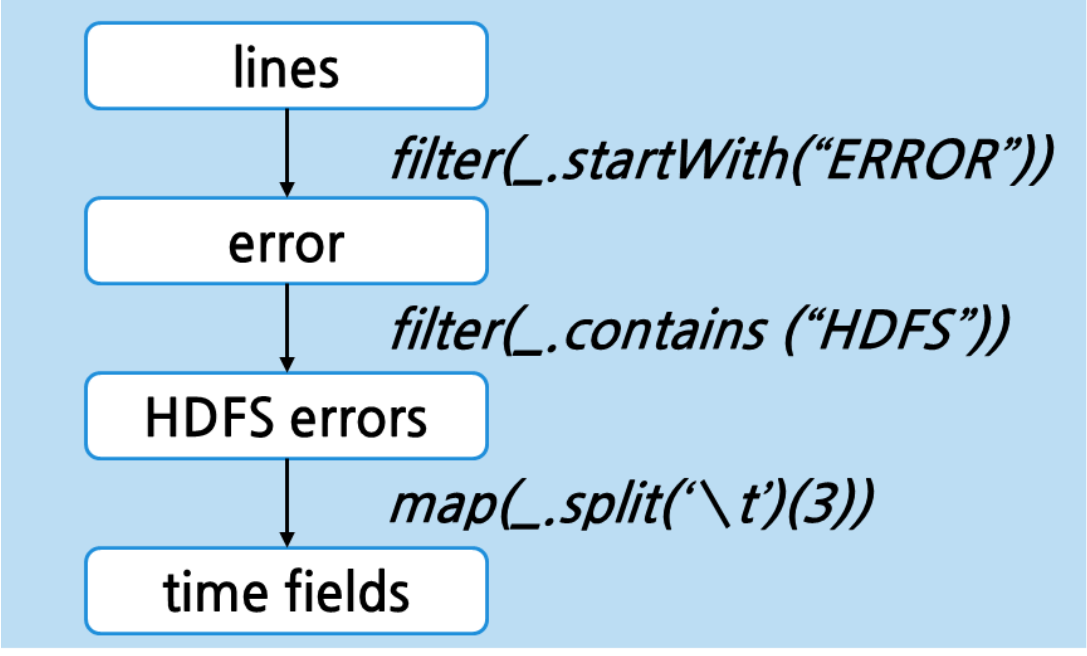
스파크 RDD

5 계보(Lineage)

RDD 연산의 순서를 기록

DAG로 표현한 것

Directed Acyclic Graph
: 직관적인 방향성 비순환 그래프



계보 예시

스파크 RDD

5 계보(Lineage)

① Fault-tolerant 확보

계보만 기록해두면 동일한 RDD를 생성할 수 있음

일부 계산 비용이 큰 RDD는 디스크에 Check Pointing함

② Lazy Execution 가능

변환 연산을 읽어 들일 때는 단순히 계보만 생성

액션 연산을 읽어 들일 때 생성된 계보를 실행함

스파크 RDD

5 계보(Lineage)

3 작업 스케줄링에 활용

✓ 일정 범위의 계보가 그려진 상태

미리 계산해서 작업 분산이 가능해짐