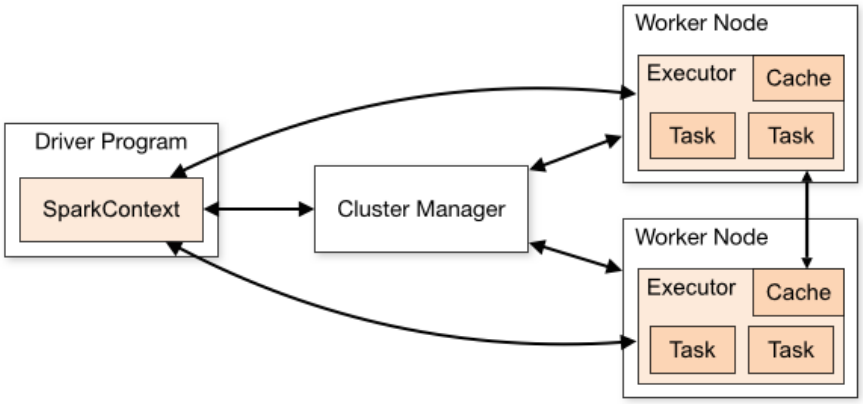
Resilient Distributed Datasets

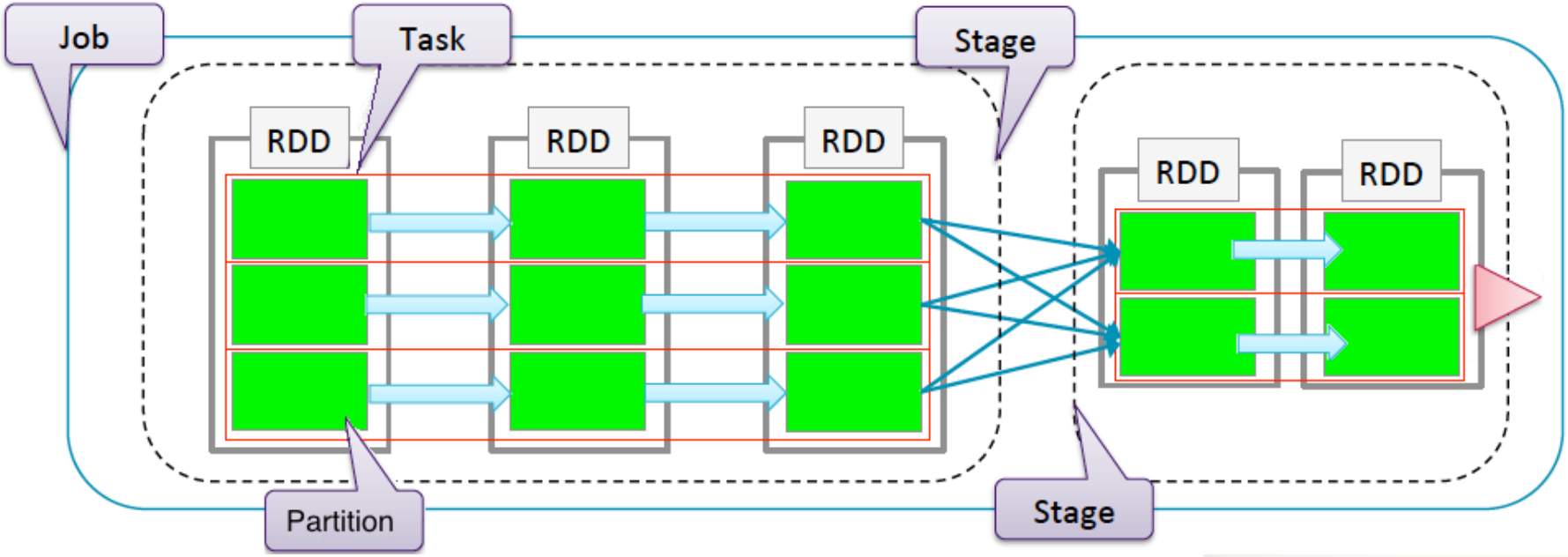
RDD

- 스파크 전용 분산 컬렉션으로 스파크의 기본 추상화 객체

- 특성
  - 불변성(immutable) → 읽기 전용(readonly)
    - » 한 번 생성된 RDD는 절대 변경할 수 없음
  - 복원성(resilient) → 장애 내성
    - » 노드에 장애가 발생해서 유실된 RDD를 원래대로 복구 가능
    - » 데이터셋을 만드는 데 사용된 작업 로그를 보존해서 장애 내성 제공
  - 분산(distributed) → 한 개 이상의 노드에 저장된 데이터셋
    - » 사용자에게는 위치 투명성 제공 → 일반적인 데이터셋을 다루는 것과 차이가 없음

RDD 실행 구조


Worker Node
Executor
Cache
Task
Task
Driver Program
SparkContext
Cluster Manager
Worker Node
Executor
Cache
Task
Task


Job
Task
Stage
RDD
RDD
RDD
RDD
RDD
Partition
Stage

RDD 연산

- Transformation과 Action이라는 두 종류의 연산 제공

- Transformation
  - RDD의 데이터를 조작해 새로운 RDD 생성 (filter, map 등)
  - Action 연산이 호출될 때까지 실행되지 않음

- Action
  - 연산을 호출한 프로그램으로 계산 결과를 변환하거나 RDD 요소에 특정 작업을 수행하기 위해 실제 계산을 시작하는 역할 (count, foreach 등)

- Action 연산이 호출되면 스파크는 RDD의 계보(lineage)를 살펴보고 이를 바탕으로 연산 그래프를 작성해서 최적화된 방식으로 연산 수행

RDD 연산 (계속)

- 데이터 타입에 따라 사용할 수 있는 연산에 차이 발생
  - 예를 들어 reduceByKey 메서드는 Key ~ Value 형태의 데이터에 대해서만 사용 가능
  - 종류

연산	설명
PairRDD Functions	Key ~ Value 형태로 구성된 데이터 대한 연산
OrderedRDD Functions	Key ~ Value 형태의 데이터 중 Key가 정렬 가능한 데이터에 대한 연산
DoubleRDD Functions	Double 형식의 데이터를 위한 연산
SequenceFile RDDFunctions	하둡의 시퀀스 파일을 다루기 위한 연산

  - 스칼라 언어로 코드를 작성할 때는 클래스 타입을 명시적으로 지정할 필요 없음

스파크 컨텍스트

- 스파크 애플리케이션과 클러스터 연결을 관리하는 객체

- 모든 스파크 애플리케이션은 반드시 스파크 컨텍스트를 생성

```
val conf = new SparkConf().setAppName(appName).setMaster(master)
new SparkContext(conf)
```

  - 마스터 서버의 정보와 애플리케이션 이름은 필수 항목

- 스파크 셸에서는 자동 생성되어 바로 사용 가능
  - sc 변수 또는 spark.sparkContext 변수로 참조

- 스파크 컨텍스트를 통해 RDD 생성

RDD 생성

- 드라이버 프로그램의 컬렉션 객체 사용

```
val rdd1 = sc.parallelize(List("a", "b", "c", "d", "e"), 2)
```

- 파일 또는 데이터베이스 등 외부 데이터 사용

```
val rdd1 = sc.textFile("file:///home/sparkdev/apps/spark/README.md")
```

RDD 주요 연산

연산	설명
collect	RDD의 모든 요소를 모아서 배열로 반환
count	RDD를 구성하는 전체 요소 개수 반환
map	RDD의 모든 원소에 지정된 함수를 적용한 뒤 그 결과로 구성된 새로운 RDD 반환
flatMap	map과 유사하게 동작하지만 결과를 1차원 요소로 풀어서 반환
mapPartitions	map, flatMap은 요소단위로 처리하지만 이 함수는 파티션 단위로 처리
mapPartitionsWithIndex	mapPartitions의 결과에 추가로 해당 파티션의 인덱스 정보도 함께 반환
mapValues	RDD의 요소가 Key ~ Value 쌍일 때 지정된 함수를 Value에만 적용해서 그 결과를 새로운 RDD로 반환
flatMapValues	map에 대한 flatMap과 비슷한 역할
zip	서로 다른 두 RDD에서 같은 인덱스의 값을 Key ~ Value 형식으로 묶어서 반환
zipPartitions	zip 연산을 파티션 단위로 수행

RDD 주요 연산

연산	설명
groupBy	RDD 요소를 일정한 기준에 따라 여러 개의 그룹으로 나누고 이 그룹으로 구성된 새로운 RDD 반환
groupByKey	Key ~ Value 형식의 RDD에 대해 키를 기준으로 그룹으로 나누고 그룹으로 구성된 새로운 RDD 반환
cogroup	Key ~ Value 형식의 여러 RDD에서 키를 기준으로 그룹으로 나누고 그룹으로 구성된 새로운 RDD 반환
distinct	중복을 제외한 요소로 구성된 RDD 반환
cartesian	두 RDD의 각 요소에 대한 카테시안 곱의 결과를 요소로 하는 RDD 반환
subtract	rdd1.subtract(rdd2)에서 rdd1에만 포함된 요소로 구성된 RDD 반환
union	두 RDD 중 어느 한 곳에라도 포함된 모든 요소로 구성된 RDD 반환
intersection	두 RDD에 동시에 모두 포함된 요소로 구성된 RDD 반환
join	Key ~ Value 형식의 두 RDD에서 같은 키를 가지고 있는 요소를 모아서 그룹을 만들고 이 결과로 구성된 RDD 반환

RDD 주요 연산

연산	설명
leftOuterJoin rightOuterJoin	join을 수행할 때 어느 한 쪽 RDD의 요소는 모두 포함하도록 처리
subtractByKey	Key ~ Value 형식의 두 RDD에서 rdd1.subtractByKey(rdd2)는 rdd1에만 포함된 Key에 해당하는 요소로 구성된 RDD 반환
reduceByKey	Key ~ Value형식의 RDD에서 같은 Key를 가진 값들을 병합해서 Key ~ Value 형식의 새 RDD 반환
foldByKey	reduceByKey와 비슷하지만 초기값을 지정 가능
combineByKey	reduceByKey, foldByKey와 비슷하지만 병합 과정에서 자료형의 변환 가능
aggregateByKey	combineByKey의 특수한 경우로 초기값 지정 가능
pipe	데이터 처리 과정에서 외부 프로세스 활용
coalesce, repartion	RDD 파티션의 개수 변경. coalesce는 감소만 가능
repartitionAndSortWithinPartitions	기준에 따라 파티션을 분할하고 각 파티션별로 포함된 요소를 정렬해서 결과로 구성된 RDD 반환

RDD 주요 연산

연산	설명
partitionBy	Key ~ Value 형식의 RDD에 대해 새로운 파티션 객체를 제공해서 다시 파티션된 RDD 반환
filter	RDD 요소 중에서 조건에 맞는 요소만 뽑아서 만든 RDD 반환
sortByKey	Key ~ Value 형식의 RDD에서 Key를 기준으로 정렬된 RDD 반환
keys, values	Key ~ Value 형식의 RDD에서 Key 또는 Value로 구성된 RDD 반환
sample	RDD에서 샘플을 추출해서 만든 RDD 반환
first	RDD의 첫 번째 요소 반환
take	RDD의 첫 번째 요소부터 지정된 개수의 요소를 뽑아서 만든 RDD 반환
takeSample	RDD에서 지정된 개수의 샘플을 뽑아서 만든 컬렉션 반환
collect	RDD의 모든 요소를 컬렉션으로 반환
count	RDD에 포함된 요소의 개수 반환

RDD 주요 연산

연산	설명
reduce	RDD의 모든 요소를 하나의 값으로 병합한 결과 반환
fold	reduce와 같은 기능을 수행하지만 초깃값 지정 가능
aggregate	reduce, fold와 같지만 결과 값의 형식이 입력 값의 형식과 다를 수 있음
sum	RDD를 구성하는 모든 요소의 자료형이 숫자형일 경우 전체 요소의 합을 반환
foreach	RDD의 모든 요소에 대해 지정된 함수 적용
foreachPartition	RDD의 파티션 단위로 지정된 함수 적용
toDebugString	RDD의 세부 정보 출력
cache, persist	첫 액션을 실행한 후 RDD 정보를 메모리 또는 디스크에 저장 → 다음 액션 수행할 때 재사용 cache는 메모리에만 저장 가능
unpersist	캐시 설정 취소. 캐시 데이터 제거
partitions	파티션 정보가 담긴 배열 반환

RDD 주요 연산

연산	설명
reduce	RDD의 모든 요소를 하나의 값으로 병합한 결과 반환
fold	reduce와 같은 기능을 수행하지만 초깃값 지정 가능
aggregate	reduce, fold와 같지만 결과 값의 형식이 입력 값의 형식과 다를 수 있음
sum	RDD를 구성하는 모든 요소의 자료형이 숫자형일 경우 전체 요소의 합을 반환
foreach	RDD의 모든 요소에 대해 지정된 함수 적용
foreachPartition	RDD의 파티션 단위로 지정된 함수 적용
toDebugString	RDD의 세부 정보 출력
cache, persist	첫 액션을 실행한 후 RDD 정보를 메모리 또는 디스크에 저장 → 다음 액션 수행할 때 재사용 cache는 메모리에만 저장 가능
unpersist	캐시 설정 취소. 캐시 데이터 제거
partitions	파티션 정보가 담긴 배열 반환