Spark

> https://www.youtube.com/watch?v=rjJ54qtOjW4&list=PL9mhQYIlKEhf23_3QIqQvsa_06CyTZGdl
>
> https://team-platform.tistory.com/49

1. MR / Spark

맵리듀스와 스파크의 차이란?

맵리듀스의 경우에는 데이터를 HDFS에서 읽고 쓰고 다시 읽고 쓰고를 반복한다. Disk I/O가 빈번하다.

스파크에서는 메모리에서 연산을 계속해서 진행하는 방식이다. 메모리에서 연산하기때문에 Disk I/O를 줄이고 네트워크 데이터 전송에도 시간을 단축시킬 수 있다.

맵리듀스와 Spark의 RDD는 low-level API이다. low-level API는 좀더 세분화된 제어가 가능하다는 것이 장점이다. Spark는 high-level API로 dataset, dataframe이 있다 이는 스칼라언어로 더 간결하게 표현할 수 있다.

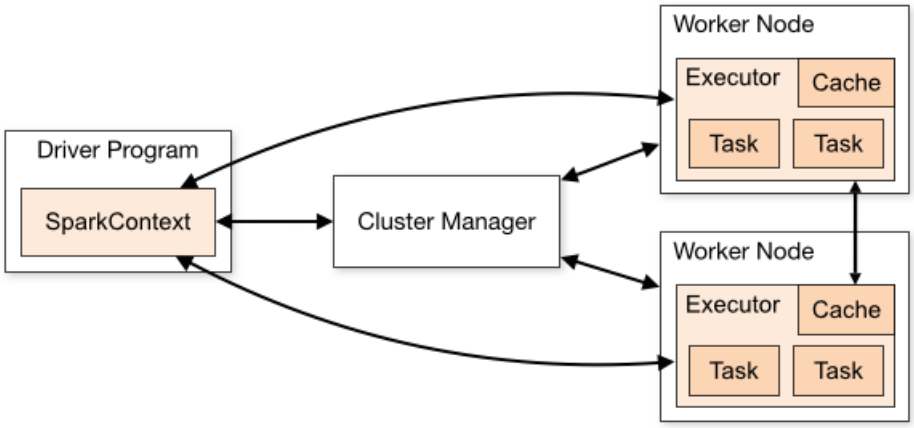
스파크는 맵리듀스보다 더 적은 노드로 더 적은 디스크 입출력으로 더 빠르게 데이터 처리할 수 있는 장점이 있다.

2. 스파크 역할

- 그래프 분석
- 실시간 스트림 데이터 처리
- SQL
- 머신러닝
- etc

3. 스파크 기본 아키텍쳐

클라이언트가 클러스터 매니저에게 사용할 executor의 개수(사용 클러스터 수, 코어 수, 메모리 수)를 요청한다.



3-1. Spark Language APIs

JVM안에있는 spark session에서는 python, R 등 다양한 언어를 지원하고 있다.

3-2. DataFrame

컴퓨터에 구조가 정의되어있는 데이터 테이블 (스프레드시트)를 분산하여 서버 클러스터에 저장되어있는 단위를 DataFrame라고 한다. 이는 스키마가 정의되어있는 데이터 구조라고 할 수 있다. (스키마 구조가 정의되어있지 않은 데이터를 RDD도 있다)

CSV파일 -> DataFrame -> Array

4. 스파크 활용 예시

1. 스트리밍 데이터 처리
2. 머신러닝
3. 그래프 데이터 분석
4. etc

5. 데이터처리

- Batch Processing : MR

- Stream Processing : Strom

  바로바로 처리되야하기때문에 데이터가 간단해야한다.

- Micro-Batching : Spark Streaming

사실 스파크는 결국 Micro-Batching에 더 가깝다. 짧게짧게 배치 프로세싱을 하는 것.

5-1. Spark Streaming 과정

1. input data가 들어온다.
2. Spark Streaming에서 정한 작은 batch 사이즈로 쪼갠다.
3. 스파크 엔진이 배치사이즈별로 처리 한다.

input 데이터를 Time별로 쪼개는 방식을 Dstream이라고한다. 각 Dstream 별로 스파크 연산을 한다. Dstream으로 쪼개진 데이터 집합들을 RDD라고 한다.

- RDD
- Window

데이터를 1초단위로 가져온다하면 RDD는 1초동안 생긴 집합니다. RDD를 1분단위로 묶으면 그것이 Window가 되는 것이다. (Window size=1분)

5-2. Checkpointing

실시간으로 처리(인메모리로 처리하다가) 문제 발생시 생성된 중간 결과물들을 저장하는 시점이다. 체크포인트 간격이 커져버리면 복구시간이나 재연산시간이 커지는 단점이 있다.

6. Spark 2.0 : Structured Streaming

스키마 있는 데이터를 SQL 쿼리로 Stream 처리하는 것.

RDD로 개발하면 진짜 내가 직접 데이터를 세심하게 핸들링하는 것이지만 데이터 프레임은 데이터 스프레드시트같이 스키마가 있는 데이터이기때문에 간편한 핸들링이 가능하다.

데이터프레임으로 스트림 처리하는 것을 Structured Streaming이다.

데이터프레임의 경우에는 Python, Scala의 성능차이가 많이 차이가 나지 않는다. 하지만 RDD에서는 Python과 Scala 성능차이가 크다.

7. 스파크 Language 차이

스파크는 Scala 언어로 개발이 되어있다. python이랑 스칼라랑 API개발이 거의 비슷하다. 하지만 R은 API가 아직 부족하다.

Scala랑 Python이랑 성능비교한 논문을 보았다.

> Journal of Korea Multimedia Society Vol. 23, No. 2, February 2020(pp. 241-246) https://doi.org/10.9717/kmms.2020.23.2.241
>
> Spark 기반에서 Python과 Scala API의 성능 비교 분석 지경엽†, 권영미††

각 동일한 CSV 파일 데이터로 mapreduce를 했을때 worker node의 수행 시간을 비교한 것이다.

결론은 Scala가 Python보다 더 좋은 성능을 가져왔다. RDD가 JVM에 저장되기 때문에 JVM위에서 동작하는 Scala가 Python보다 더 좋은 성능을 가져왔다.

JVM에서 실행이 된다면 자바와는 차이점은 무엇인가?

자바와 스칼라 모두 정적 타입핑 언어이다. 타입을 알려줘야하는 언어이지만 스칼라는 때에따라 타입을 생략하고 컴파일러가 타입추론을 해주는 경우도 있다. 그렇기때문에 코드가 더 간결해질 수 있다. 즉, 코드의 간결성이 스칼라 언어의 장점이라고 할 수 있다.