Spark - Week 3

Created	@2022년 3월 20일 오후 8:32
Tags	

스파크 GroupByKey, ReduceByKey 비교하기!

GroupByKey

- 파티션 내부에서 같은 키값을 가진 것끼리 묶어서 전체 키 그룹에 대해서 카운트를 함
- 그룹 전체가 메모리에 올라가게 됨!

ReduceByKey

- 파티션 내부에서 RDD를 같은 key값을 가진 것끼리 묶어서 카운트를 진행하고, 이 파티션들을 다시 묶어서 카운트를 진행함
- 리듀스 연산의 결과로 또다른 리듀스 연산을 진행함
- reduce 연산 종류
  - sum
  - max
  - min
- 어떤 Value 집합에 대해서 reduce 연산을 할 수 있는 필요조건
  - 두 입력값과 출력값의 타입이 같다

    $f(V, V) \rightarrow V$
  - 연산 간 교환법칙이 성립한다

    $f(V_1, V_2) = f(V_2, V_1)$
  - 연산 간 결합법칙이 성립한다

    $f(V_1, (V_2, V_3)) = f((V_1, V_2), V_3)$

둘중에 뭐를 쓰는게 좋은가?

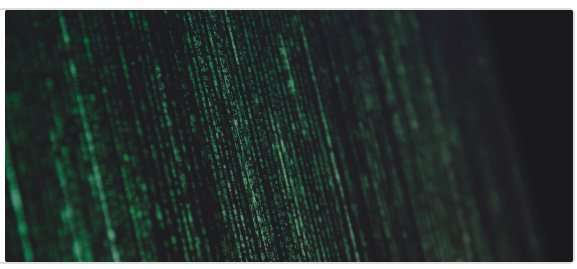
- 다음과 같은 경우에는 `groupByKey`를 쓰는 것이 좋다!
  - 그룹 전체를 메모리에 올려서 복잡한 계산을 수행하거나 여러 번 순회하며 계산해야 하는 경우
  - 그룹의 최대 크기가 부담없이 메모리에 올라갈 정도로 작으면서, `reduceByKey`로 대체했을 때의 셔플 횟수가 `groupByKey`를 썼을 때보다 큰 경우
- 다음과 같은 경우에는 `reduceByKey`로 대체하는 것이 좋다!
  - 그룹의 최대 크기가 메모리에 올리기 부담스러울 정도로 클 경우
  - `reduceByKey` 로 대체했을 때의 셔플 횟수가 `groupByKey`를 썼을 때보다 작거나 같은 경우

참고한 사이트

Spark RDD에서 GROUP BY를 빠르게 하려면? - 리디주식회사 RIDI Corporation

Apache Spark 는 대용량 데이터의 범용 계산을 위한 분산처리 시스템입니다. 기존의 Hadoop Map-Reduce에 비해 훨씬 빠르면서도 간편하게 복잡한 데이터 연산을 처리할 수 있습니다. 전 세계적으로도 많은 사용자층을 가지고 있으며, 2018년 8월 현재 2.3.1 버전까지 릴리즈되었을 정도로 유지보수도 꾸준히 이루어지고 있습니다.

https://ridicorp.com/story/park-rdd-groupby/



Databricks Spark Knowledge Base

Let's look at two different ways to compute word counts, one using reduceByKey and the other using groupByKey: val words = Array("one", "two", "two", "three", "three", "three") val wordPairsRDD = sc.parallelize(words).map(word => (word, 1)) val wordCountsWithReduce = wordPairsRDD

https://databricks.gitbooks.io/databricks-spark-knowledge-base/content/best_practices/prefer_reducebykey_over_groupbykey.html