Group assignment

-RDD vs DataFrame vs SparkSQL queries-

Group2

주영훈,류재훈,오영택,김영민

수업시간에서 배운 내용을 토대로 spark 분산 처리 환경에서 data processing 형태에 따른 비교를 진행하였다.

추가적으로

orderHeader = sc.textFile("SalesLTSalesOrderHeader.txt")

customer = sc.textFile("SalesLTCustomer.txt")

이 부분에서 sc.textFile은 file을 local에서 읽는다고 명시하지 않으면 hdfs에서 읽는다고 생각하기때문에 에러가 발생하여, 로컬경로로 변환하였다

orderHeader = sc.textFile("file:///home/project/group\_assignment/SalesLTSalesOrderHeader.txt")

customer = sc.textFile("file:///home/project/group\_assignment/SalesLTCustomer.txt")

1. Retrieve customer orders
   1. RDD way

Code line: 20 line (involved visualization code)

Time: CPU times: user 596 ms, sys: 180 ms, total: 776 ms, Wall time: 1.28 s

* 1. DataFrame way

Code line: 8 line

Time: CPU times: user 80 ms, sys: 16 ms, total: 96 ms, Wall time: 12.4 s

* 1. SparkSQL queries way

Code line: 3 line

Time: CPU times: user 12 ms, sys: 8 ms, total: 20 ms, Wall time: 1.46 s

1. Retrieve customer orders with addresses
   1. RDD way

Code line: 10 line

Time: CPU times: user 48 ms, sys: 12 ms, total: 60 ms, Wall time: 1.47 s

* 1. DataFrame way

Code line: 5 line

Time: CPU times: user 56 ms, sys: 16 ms, total: 72 ms, Wall time: 7.64 s

* 1. SparkSQL queries way

Code line: 3 line

Time: CPU times: user 28 ms, sys: 0 ms, total: 28 ms, Wall time: 1.99 s

1. Retrieve a list of all customers and their orders
   1. RDD way

Code line: 12 line

Time: CPU times: user 32 ms, sys: 0 ms, total: 32 ms, Wall time: 150 ms

* No visualization so, it has faster than below measure.
  1. DataFrame way

Code line: 5 line

Time: CPU times: user 64 ms, sys: 8 ms, total: 72 ms, Wall time: 2.18 s

* 1. SparkSQL queries way

Code line: 4 line

Time: CPU times: user 20 ms, sys: 12 ms, total: 32 ms, Wall time: 1.84 s

1. Retrieve a list of customers with no address
   1. RDD way

Code line: 7 line

Time: CPU times: user 24 ms, sys: 0 ms, total: 24 ms, Wall time: 251 ms

* 1. DataFrame way

Code line: 4 line

Time: CPU times: user 32 ms, sys: 4 ms, total: 36 ms, Wall time: 525 ms

* 1. SparkSQL queries way

Code line: 3 line

Time: CPU times: user 0 ms, sys: 4 ms, total: 4 ms, Wall time: 347 s

It’s all good for evaluation with time. It support Lazy evaluation.

But there’s some difference on result.

Why we got this result?

Because, RDD has only little Lazy evaluation way for analysis. But Dataframe, has both Lazy evaluation and transformation modifed processing(by Catalyst Optimizer) so It’s better than pre-analysis.

RDD note ‘how’ do I analysis. But Dataframe note ‘what’ do I analysis. So, Dataframe has optimaized analysis processing.

On code line.

Dataframe make new schema for data analysis. And do job on this. So, this making work has some code line, and takes time.

we have shortest line on SparkSQL queries way. Because it works like SQL QUERY.

When I send query on data. This query is optimized by automatically. And It’s fly to data.

Plus, it’s a few code line. so works better on time.

Result(느낀 점)

데이터 분산 처리에서 여러 방법(RDD way, DataFrame, SparkSQL query)들을 실제로 코드로 돌려보면서 각각의 방법들에 대해서 조금 더 이해할 수 있었다. 모든 방법이 어느정도 Lazy evaluation을 지원하고 있었고, dataframe 경우 transformation의 최적화의 과정도 포함하고 있어 더 속도가 빨랐다는 것을 알 수 있었다. 그리고 spark에서 sql query를 날릴 수 있는 통로역할을 하는 spark sql에 대해서 코드 라인과 성능 차이에 대해서 비교해 볼 수 있었다. Line & time을 기준으로 보았을 때, 가장 좋았던 것은 spark sql이었다.