Hadoop In Real World is now Big Data In Real World! Read here! X

BIG DATA IN REAL WORLD

HOME COURSES REVIEWS BLOG ABOUT

What is the difference between foreach and foreachPartition in Spark?

Published by Big Data In Real World at October 4, 2021 Tags Categories

foreach() and foreachPartition() are action function and not transform function. Both functions, since they are actions, they don't return a RDD back.

Do you like us to send you a 47 page Definitive guide on Spark join algorithms? ===>

Send me the guide

foreach()

Use foreach() when you want to apply a function on every element in a RDD. But note, you are not transforming the elements in the RDD. With foreach() you are usually changing the state of something outside the RDD based on the elements in the RDD. Thereby causing side effects.

For eg. you can use foreach() to update a column in a database table for every element in RDD.

A common use case to use foreach() is to update an accumulator for every element in RDD.

```
scala> val acc = sc.longAccumulator("sample-accumulator")

scala> sc.parallelize(Array(10, 20, 30, 40)).foreach(element => acc.add(element))
```

foreachPartition() is very similar to mapPartitions() as it is also used to perform initialization once per partition as opposed to initializing something once per element in RDD.

With the below snippet we are creating a Kafka producer inside foreachPartition() and sending the every element in the RDD to Kakfa.

```
rdd.foreachPartition { //called once per partition

  partition =>
    val producer = createKafkaProducer()
    partition.foreach { //called per each element in the partition
    element => producer.send(element)
  }

  producer.close()

}
```

Let's say our RDD has 5 partitions and 10 elements in each partition. So a total of 50 elements in total. At execution each partition will be processed by a task. Each task gets executed on worker node.

With the above code snippet, foreachPartition will be called 5 times, once per task/partition. So each task will create kafkaProducer. Inside each partition, foreach function will be called for every element in the partition. So in total it will called 50 times.

Conclusion

Again, note that both foreach and foreachParitition are used for operation which causes side effects and they are intended to change the state of the something (DB, accumulator, Kafka etc.) other than the RDD. Both functions are not designed to transform the RDD as they are not transformation functions, they are actions.

If you are trying to transform the RDD, refer the post related to map and mapPartitions.

Big Data In Real World

We are a group of Big Data engineers who are passionate about Big Data and related Big Data technologies. We have designed, developed, deployed and maintained Big Data applications ranging from batch to real time streaming big data platforms. We have seen a wide range of real world big data problems, implemented some innovative and complex (or simple, depending on how you look at it) solutions.

Related posts

June 1, 2025

Sunset: Spark Developer In Real World cluster

Read more

October 8, 2023

How to kill a running Spark application?

Read more

September 25, 2023

What is the default number of executors in Spark?

Read more

Comments are closed.