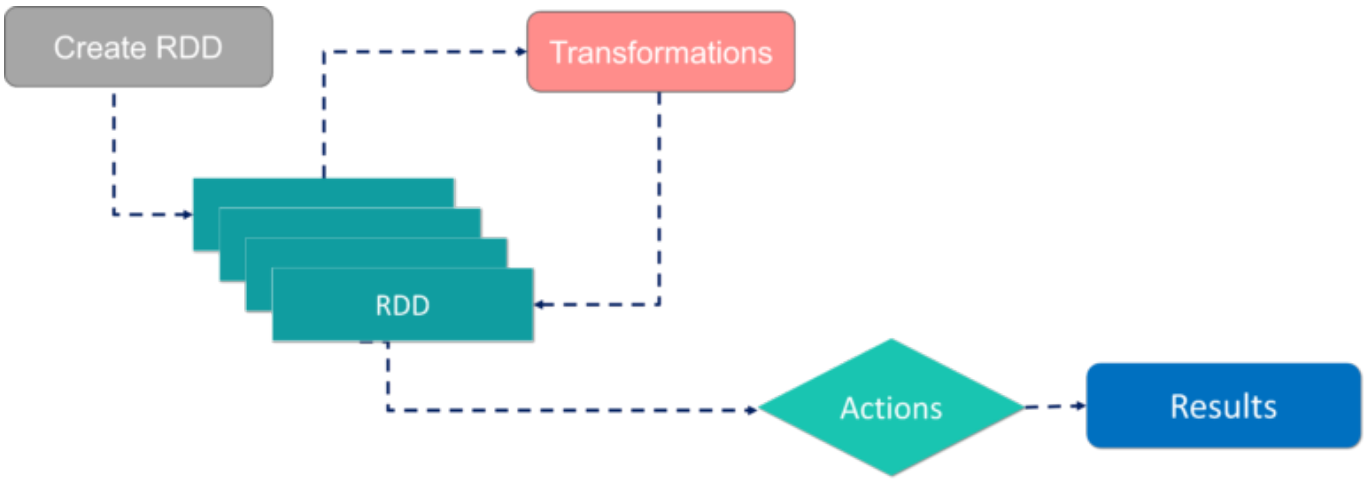

image source — quora.com

Why is RDD immutable?

Amit Singh Rathore Follow
Apr 26, 2020 · 2 min read

What benefit do we get out of it?

Before I go further, what is RDD? — *RDD is not a collection of Data. RDD is an abstraction to create a collection of data. It is just a set of description or metadata which will, in turn, when acted upon, give you a collection of data.*

Now the why? First thing, Spark is written in Scala, which supports various aspects of functional programming like currying, lazy evaluation, and so on. In my opinion Spark developer might have decided to leverage this aspect and they might have decided that they need an abstraction that will be computed in a deterministic way and should be able to support concurrent consumption. RDD's immutability fits right in the slot here. Spark speeds up performance by using in-memory computations. It's very likely that you will want your in-memory "stuff" to be immutable since it will remove the need for the frequent cache invalidation. Again RDDs immutability fits in here. Multiple threads accessing the same data and operating on that, immutability removes any requirements of sync up between nodes in a distributed environment.

Lineage: Just think if RDDs are not immutable. Will we be able to deterministically regenerate the previous step once we encounter failure? — **No.**

I guess we have enough arguments why RDDs are immutable.

Happy Learning!

Spark Spark Rdd