RDD

Broadcast Joins

Objective

Implement the broadcasting technique on RDDs

Quick reminder

- used for joining a large "table" with a small "table", e.g. a lookup table
- copy the small "table" entirely, on all the executors
- no shuffle
- blazing fast

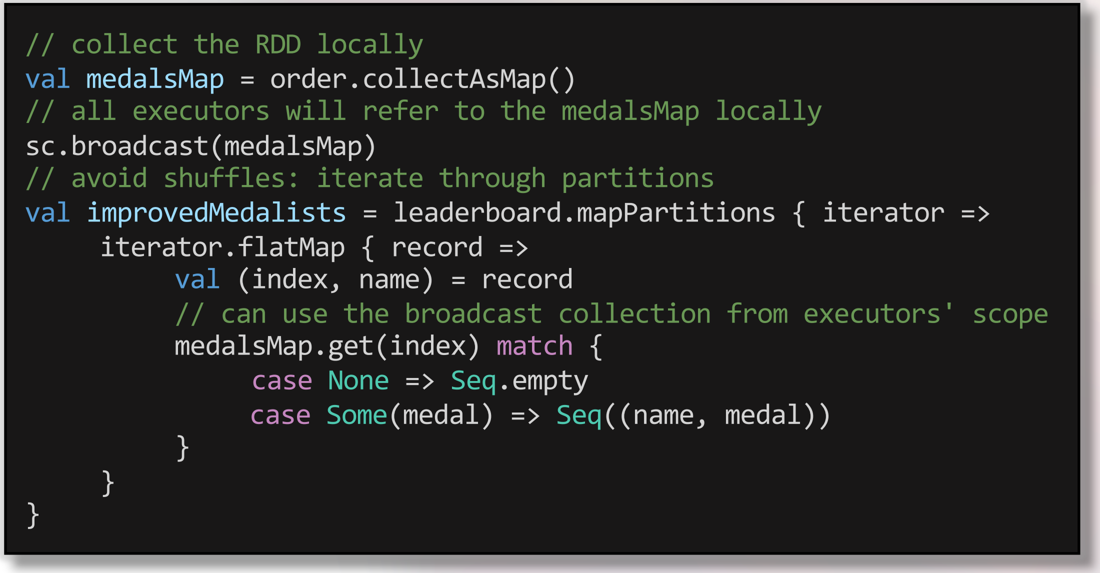
To Remember

Broadcasting is useful when one RDD is small

- send it to all executors
- no shuffles needed

Need to do broadcasting ourselves

- collect the small RDD locally
- call broadcast on the SparkContext
- mapPartitions on the big RDD
- use the collection locally in executors



```scala
// collect the RDD locally
val medalsMap = order.collectAsMap()
// all executors will refer to the medalsMap locally
sc.broadcast(medalsMap)
// avoid shuffles: iterate through partitions
val improvedMedalists = leaderboard.mapPartitions { iterator =>
     iterator.flatMap { record =>
          val (index, name) = record
          // can use the broadcast collection from executors' scope
          medalsMap.get(index) match {
               case None => Seq.empty
               case Some(medal) => Seq((name, medal))
          }
     }
}
```

Spark rocks