Himanshu Goyal

Spark Assignment 5

Spark provides two limited types of shared variables for two common usage patterns:

1) Broadcast variables
2) Accumulators.

1) Broadcast variables-
   We will use these variables if we need to share them throughout the cluster. They are immutable meaning we cannot change them or we can call them secured read only variables. These variables have memory constraint and should not be relatively larger.

   The example of such variables can be like store meta data about our program where we don't need to store much information.

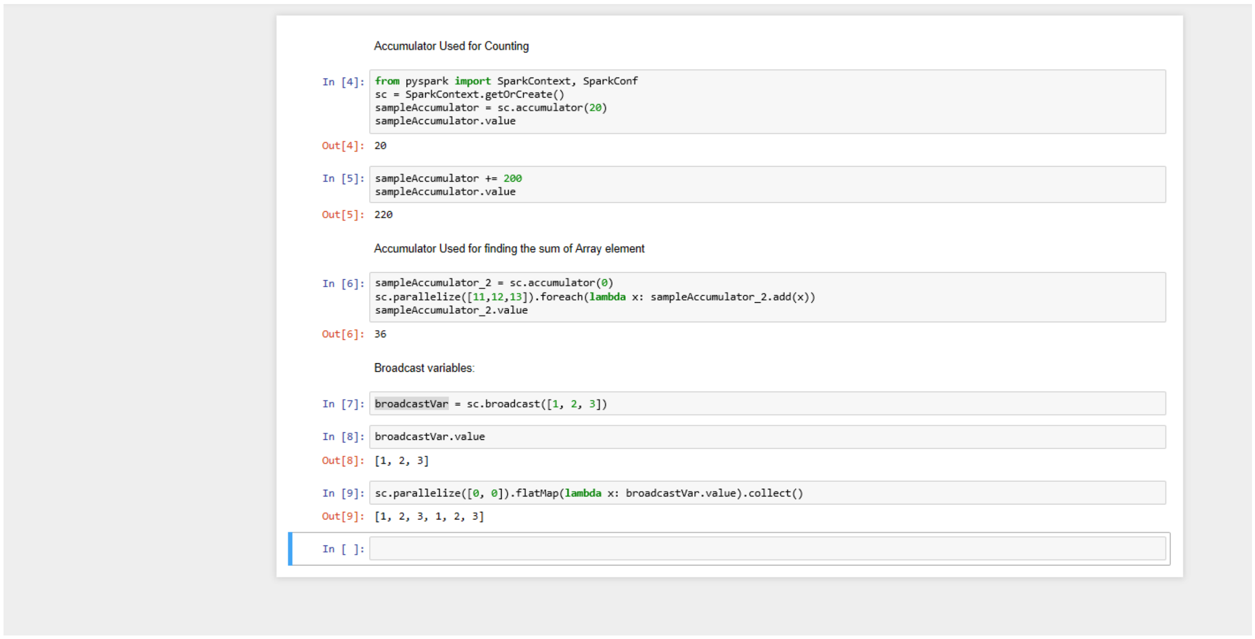
   "One of the really cool thing about broadcast variables is that in Spark, they're handled by a torrent-like protocol. What happens is that the nodes in the cluster all try to distribute the variable as quickly and efficiently as possible by downloading what they can and uploading what they can. This makes them much faster than one node having to try and do everything and push the data to all nodes." [1]

   Broadcast variables are created by wrapping with SparkContext.broadcast function as shown in the following Scala code. [2]x

2) Accumulators
   Accumulators as a global counter variable where each node of the cluster can write values in to it. These can be used while reading log lines. Spark natively supports accumulators of numeric types, and programmers can add support for new types. We can have named or unnamed accumulators. An accumulator is created from an initial value v by calling SparkContext.accumulator(v). Spark guarantees that each task's update to the accumulator will only be applied once,
   Accumulator is simply for counter purpose it's value can be read by Driver only. [3][4]

Reference:
[1]: http://www.sparktutorials.net/Spark+Broadcast+Variables+-+What+are+they+and+how+do+I+use+them
[2]: http://www.sparktutorials.net/Spark+Broadcast+Variables+-+What+are+they+and+how+do+I+use+them
[3]: https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html

[4]: https://github.com/vaquarkhan/vaquarkhan/wiki/What-are-broadcast-variables-and-accumulators-inApache-Spark%3F-Answer-Request



Accumulator Used for Counting

```
In [4]: from pyspark import SparkContext, SparkConf
        sc = SparkContext.getOrCreate()
        sampleAccumulator = sc.accumulator(20)
        sampleAccumulator.value

Out[4]: 20

In [5]: sampleAccumulator += 200
        sampleAccumulator.value

Out[5]: 220
```

Accumulator Used for finding the sum of Array element

```
In [6]: sampleAccumulator_2 = sc.accumulator(0)
        sc.parallelize([11,12,13]).foreach(lambda x: sampleAccumulator_2.add(x))
        sampleAccumulator_2.value

Out[6]: 36
```

Broadcast variables:

```
In [7]: broadcastVar = sc.broadcast([1, 2, 3])

In [8]: broadcastVar.value

Out[8]: [1, 2, 3]

In [9]: sc.parallelize([0, 0]).flatMap(lambda x: broadcastVar.value).collect()

Out[9]: [1, 2, 3, 1, 2, 3]

In [ ]:
```