

## Problem Statement

Count the number of blank lines in a text file, by using accumulators

### Sample file :

Hello World

It's a sunny day

<blank\_line>

When will it rain?

Will it rain today!

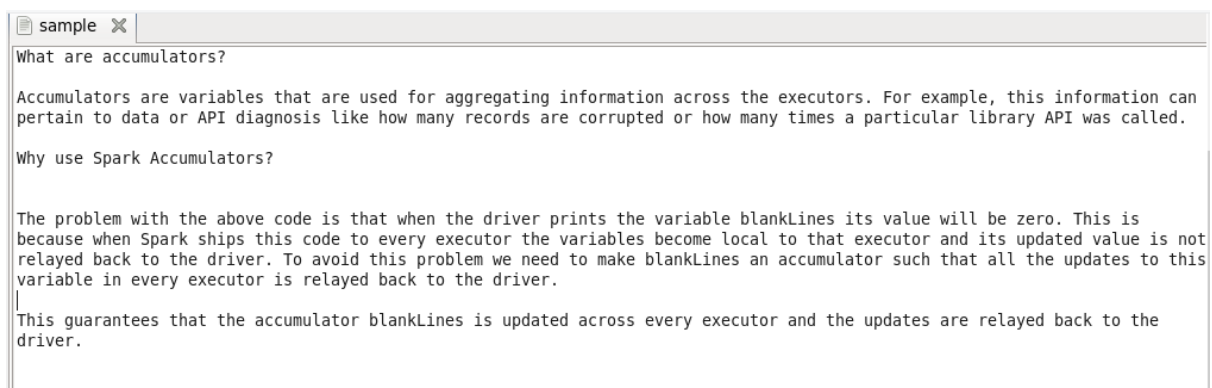
## Solution:

Accumulators give us a simpler way of aggregating data. Accumulators help in keeping aggregate variables in being across every executor and that these updates are relayed back to the driver.

To solve the given problem, follow the below steps:

- Begin by reading the data file as a text file from the local FS using the spark context object `sc`. file

Here I am using sample as a text file.



```
sample X
What are accumulators?

Accumulators are variables that are used for aggregating information across the executors. For example, this information can
pertain to data or API diagnosis like how many records are corrupted or how many times a particular library API was called.

Why use Spark Accumulators?

The problem with the above code is that when the driver prints the variable blankLines its value will be zero. This is
because when Spark ships this code to every executor the variables become local to that executor and its updated value is not
relayed back to the driver. To avoid this problem we need to make blankLines an accumulator such that all the updates to this
variable in every executor is relayed back to the driver.

This guarantees that the accumulator blankLines is updated across every executor and the updates are relayed back to the
driver.
```

- Then create two variables `blankLines` and `validLines` as **accumulators** with value 0 using `sc`.
- To compute the number of blank and non-blank lines, each line of the file is checked to see if the count elements in the line is 0, If so then it means that the line is blank and we increment the accumulator **blankLines**. If not then it means that the line is not blank and the accumulator **validLines** is incremented.
- Finally, print the value of both the accumulators.

## Code:

```
scala> val file = sc.textFile("/home/acadgild/sample")
file: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:27

scala>

scala> val blankLines = sc.accumulator(0)
blankLines: org.apache.spark.Accumulator[Int] = 0

scala>

scala> val validLines = sc.accumulator(0)
validLines: org.apache.spark.Accumulator[Int] = 0

scala>
scala> val count = file.foreach {
  | line => if(line.length() == 0)
  |   blankLines +=1
  |   else
  |   validLines +=1
  | }
count: Unit = ()
```

---

## Output:

```
scala> println("\n\n Blank Lines in file sample is : " + blankLines.value + "\n")

Blank Lines in file sample is : 5

scala> println("\n\n Non-Blank Lines in file sample is : " + validLines.value + "\n")

Non-Blank Lines in file sample is : 5

scala> █
```

---