|  |
| --- |
| /\* |
|  | \* Application to count the number of blank lines and spaces in file |
|  | \* |
|  | \* Sample data: |
|  | \* Hello World |
|  | It is a sunny day |
|  |  |
|  | When will it rain? |
|  | Will it rain today! |
|  | \*/ |
|  |  |
|  | package assignment |
|  |  |
|  | import org.apache.spark.SparkConf |
|  | import org.apache.spark.SparkContext |
|  | import org.apache.spark.rdd.RDD |
|  |  |
|  | object AccumulatorsAssignment { |
|  | def main(args: Array[String]): Unit = { |
|  |  |
|  | //specify the configuration required for spark app to run , This will run in local mode by setting master as local |
|  | val config = new SparkConf().setAppName("Accumulators Demo").setMaster("local") |
|  |  |
|  | //SparkContext object for creation of rdd's |
|  | val sc = new SparkContext(config) |
|  |  |
|  | //create an rdd from textFile |
|  | val file = sc.textFile("/home/acadgild/Spark/assignment/accumulators") |
|  |  |
|  | // call function which uses accumulators for calculating number of blank lines and spaces in file |
|  | usingAccumulator(sc,file) |
|  | } |
|  |  |
|  | /\* |
|  | \* function which uses accumulators for calculating number of blank lines and spaces in file |
|  | \*/ |
|  | def usingAccumulator(sc:SparkContext,rdd:RDD[String]){ |
|  |  |
|  | //define accumulator for space calculation |
|  | val spaceAccumulator = sc.accumulator(0,"space\_accum") |
|  |  |
|  | //define accumulator for blank lines calculation |
|  | val blankLineAccumulator = sc.accumulator(0,"space\_accum") |
|  |  |
|  | //convert the strings to stream of chars and check it equals to space and increment accumulator |
|  | rdd.foreach{line => {line.toStream.foreach(char => if( char==' ') spaceAccumulator += 1) }} |
|  |  |
|  | //convert line is empty and increment accumulator |
|  | rdd.foreach(line => {if (line.isEmpty()) blankLineAccumulator += 1 }) |
|  |  |
|  | //print the values of accumulators |
|  | println("Number of spaces :" + {spaceAccumulator.value}) |
|  |  |
|  | println("Number of Blank lines :" + {blankLineAccumulator.value}) |
|  | } |
|  | } |

