

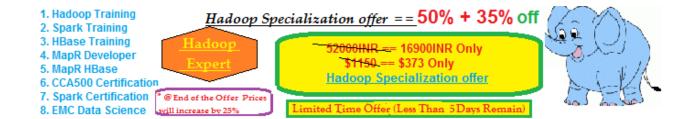


SPARK API HANDSON LAB USE BROADCAST VARIABLE, FILTER FUNCTION AND SAVE OUTPUT TO HDFS AS TEXT FILE

By www.HadoopExam.com

Note: These instructions should be used with the HadoopExam Apache Spark: Professional Trainings.

Where it is executed and you can do hands on with trainer.



Cloudera CCA175 (Hadoop and Spark Developer Hands-on Certification available with total 75 solved problem scenarios. Click for More Detail)





Problem Scenario: You have given following two files.

- 1. **Content.txt**: Contain a huge text file containing space separated words.
- 2. **Remove.txt**: Ignore/filter all the words given in this file (Comma Separated).

Write a Spark program which reads the Content.txt file and load as an RDD, remove all the words from a broadcast variables (which is loaded as an RDD of words from Remove.txt). And count the occurrence of the each word and save it as a text file in HDFS.

Content.txt	Remove.txt
Hello this is HadoopExam.com	Hello, is, this, the
This is QuickTechie.com	
Apache Spark Training	
This is Spark Learning Session	
Spark is faster than MapReduce	

Solution:

<u>Step 1:</u> Create both the files in hdfs in a directory called spark2 (We will do using Hue). However, you can first create in local filesystem and then upload it to hdfs.

Step 2: Load the Content.txt file

val content = sc.textFile("spark2/Content.txt")

Step 3: Load the Remove.txt file

val remove = sc.textFile("spark2/Remove.txt")

<u>Step 4:</u> Create an RDD from remove, however, there is a possibility each word could have trailing spaces, remove those whitespaces as well. We have used two functions here flatMap, map and trim.

val removeRDD= remove.flatMap(x=> x.split(",")).map(word=>word.trim)

Step 5: Broadcast the variable, which you want to ignore

val bRemove = sc.broadcast(removeRDD.collect().toList) // It should be array of Strings

Step 6: Split the content RDD, so we can have Array of String.

val words = content.flatMap(line => line.split(" "))

Step 7: Filter the RDD, so it can have only content which are not present in "Broadcast Variable".

val filtered = words.filter{case (word) => !bRemove.value.contains(word)}

Step 8: Create a PairRDD, so we can have (word,1) tuple or PairRDD.

val pairRDD = filtered.map(word => (word,1))

Step 9: Now do the word count on PairRDD.

val wordCount = pairRDD.reduceByKey(+)





Step 10: Save the output as a Text file.

wordCount.saveAsTextFile("spark2/result.txt")

Spark Professional Training with Hands on Lab Session

http://www.HadoopExam.com

HadoopExam Learning Resource provides the following material for the Advanced Technologies. Please visit www.HadoopExam.com for more detail this is just a few products from portfolio.

Price start for training with Just \$79/3500INR



HandsOn Session

+ Certification

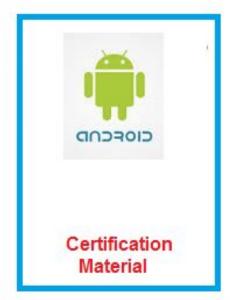
Material



+ Certification

Material









Certification Material



Certification Material



Certification Material



