Respostas das questões

**Qual o objetivo do comando cache em Spark?**

R: O comando cache serve em casos que um dado precisa ser consultado diversas vezes, o dado é armazenado em memória para agilizar o processo, pois não será necessário refazer o processamento. Mas precisa ser usado com cuidado, pois pode prejudicar a performance e onerando a memória de forma indevida.

**O mesmo código implementado em Spark é normalmente mais rápido que a implementação equivalente em MapReduce. Por quê?**

R: As tarefas executadas no Spark são processadas em memoria, enquanto MapReduce utiliza os motores de busca baseados em discos.

**Qual é a função do SparkContext ?**

R: SparkContext representa uma conexão com um cluster Spark, sem ela não é possível criar RDD e utilizar outras funções dentro do ambiente Spark.

**Explique com suas palavras o que é Resilient Distributed Datasets (RDD).**

R: RDD é um estrutura de dados fundamental para o Spark, consiste em particionar conjuntos de dados para leitura, que suporta processamento em memória, sendo totalmente tolerante a falhas, pois é capaz de refazer partições com erros.

**GroupByKey é menos eficiente que reduceByKey em grandes dataset. Por quê?**

R: Sim, pois em grandes datasets GroupByKey forma uma enorme combinação de elementos, diferente do reduceByKey que combina as saídas com uma chave comum em cada partição antes de agregar.

**Explique o que o código Scala abaixo faz:**

val textFile = sc . textFile ( "hdfs://..." )

val counts = textFile . flatMap ( line => line . split ( " " ))

. map ( word => ( word , 1 ))

. reduceByKey ( \_ + \_ )

counts . saveAsTextFile ( "hdfs://..." )

R: Inicialmente é criado um RDD com o arquivo diponivel no hdfs (“hdfs://”), depois o arquivo e dividido usando o “ espaço” como referencia e é transformado numa CHAVE-VALOR e para cada registro é atribuído o valor 1, depois os valores serão agrupados e reduzidos com a soma dos valores por chave, e após é salvo em algum diretório do HDFS.