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

Guardar um conjunto de dados em memória cache, para que possam ser acessados mais rapidamente. Isso é útil principalmente quando um certo conjunto de dados precisa ser acessado com certa frequência.

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

Porque o MapReduce apesar de possibilitar paralelismo tem a limitação de ter que gravar os resultados em memória secundária todas as vezes, enquanto o Spark possibilita trabalhar com os dados nos dois tipos de memória (e quando trabalha com memória secundária ainda assim é mais rápido)

Qual é a função do **SparkContext** ?

Definir como o spark deve acessar um determinado cluster. Ele representa a conexão com esse cluster e permite a criação de RDDs.

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

Uma coleção de elementos imutáveis e distribuídos que podem ser manipulados em paralelo e processados em diferentes nós de um cluster. É uma estrutura de dados fundamental do Spark.

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

Porque o comando GroupByKey movimenta todas as entradas chave-valor quando chamado, enquanto o reduceByKey combina os pares da mesma máquina com a mesma chave antes que os dados sejam movimentados.

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://...")
```

Conta quantas palavras há em um arquivo e depois salva este número em outro arquivo.

Para resolução das questões a documentação do spark foi consultada, além dos seguintes websites:
https://www.infoq.com/br/articles/apache-spark-introduction
https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm
https://databricks.gitbooks.io/databricks-spark-knowledge-base/content/best_practices/prefer_reducebykey_over_groupbykey.html

Questões

Responda as seguintes questões devem ser desenvolvidas em Spark utilizando a sua linguagem de preferência.

1. Número de hosts únicos.

2. O total de erros 404.
   30795

3. Os 5 URLs que mais causaram erro 404.

4. Quantidade de erros 404 por dia.

5. O total de bytes retornados.

Favor ler os comentários da classe Main.java