Nome: Oswaldo de Castilho Scanholato Neto

Qual o objetivo do comando cache em Spark?

R: O uso do comando em cache ajuda melhorar a eficiência do código. Permitindo que os resultados possam ser armazenados e reutilizados mais rápido.

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

R: É porque o spark é executado in memory enquanto que o map red é feita escrita e leitura em disco cache é o comando para armazenar resultado intermediários das rdds que serão utilizados novamente para ganho de performance ao invés de ficar executando toda vez q for chamado.

Qual é a função do SparkContext?

R:  SparkContext pode ser acessado como uma variável em um programa que para utilizar os seus recursos.

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

R:  Resilient Distributed Datasets serve para a manipulação de dados, é uma representação de um dado distribuído pelos nós do cluster que pode ser operado em paralelo. RDDs podem ser criados a partir de arquivos no HDFS ou de coleções.

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

R: ReduceByKey, os dados se combinam em cada partição, somente uma saída para chave em cada partição. Ele requer combinar todos seus valores em outros valores com exatamente do mesmo tipo.

GroupByKey, ele pode causar alguns problemas fora do disco em que os dados se enviam através de leitura na sua recopilação de redução.

val textFile = sc.**textFile**("hdfs**:**//...") val counts = textFile.**flatMap**(line => line.split(" ")) **.map**(word => (word, 1)) **.reduceByKey**(\_ + \_) counts.saveAsTextFile("hdfs://...")

R: Lê os dados armazenados em hdfs em um RDD de Strings chamado textFile.

**flatMap**: pegamos o RDD de linhas e o transformamos em um RDD de palavras

**.map:** transformamos o RDD de palavras em RDD (word, 1). Também é chamado de RDD de valor-chave

**.reduceByKey:** para cada chave (word) reduzimos todos os valores somando todos os valores juntos. Reduzir operação em geral, não no sentido de Spark ou Scala faz isso, levamos algumas listas e aplica algumas funções várias vezes ao resultado anterior e ao próximo elemento, por exemplo.

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.

R: Select count(cd\_erro) from tabela where cd\_erro = 404;

3. Os 5 URLs que mais causaram erro 404.

R: select top 5 count(cd\_erro) from tabela where cd\_erro = 404;

4. Quantidade de erros 404 por dia.

R: Select count(\*), dt\_acesso from tabela where erros = 404 group by dt\_acesso;

5. O total de bytes retornados.

R: select sum(bytes) from tabela;