1 - Qual o objetivo do comando cache em Spark?

Os acessos diretos ao disco, geram atrasos devido a repetida execução. Para o caso do RDD, o uso do comando cache, faz com que os dados que já foram pesquisados sejam acessando de forma rápida através de dados já existentes em memória. Essa Função ajuda para que as informações sejam mostradas de forma mais rápida e com menos acesso ao disco.

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

O mapreduce é executado muito utilizado pelo hadoop, no spark utilizamos o RDD que é uma estrutura de implementação voltada a dados em memória e que são utilizadas para armazenar em cache os dados existentes entre os nós de um cluster tornando o acesso os registros muito mais rápido. Em o Spark é muito mais eficiente para a execução rápida de códigos do que o equivalentes ao mapreduce.

3 - Qual é a função do SparkContext ?

O contexto do Spark configura serviços internos e estabelece uma conexão com um ambiente de execução do Spark. Um contexto do Spark é essencialmente um cliente do ambiente de execução do Spark e atua como o mestre do seu aplicativo Spark. Nas configurações do sparkcontext você passa algumas instruções como:

nome da aplicação Spark, que será apresentada na interface do cluster, e o Cluster Manager a ser utilizado.

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

Os RDDs são uma abstração de memória distribuída que permite que os programadores realizem cálculos na memória em clusters grandes de maneira tolerante a falhas. Imagine o RDD como uma tabela do banco de dados que pode guardar qualquer tipo de dado. O Spark armazena os dados do RDD em diferentes partições. Isso ajuda a reorganização computacional e a otimização no processamento das informações. É uma coleção de dados espalhados por diversos partições.

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

A diferencia é na quantidade de informações e na forma de agrupamentos dos dados,O groupby os dados são embaralhados de acordo com a chave e Nessa transformação, muitos dados desnecessários são transferidos pela rede, devido a isso pode causar problemas de falta de disco à medida que os dados são enviados pela rede e coletados nos trabalhadores de redução.

No reduce antes de embaralhar os dados, os pares na mesma máquina com a mesma chave são combinados. Os dados são combinados em cada partição, apenas uma saída para uma chave em cada partição a ser enviada pela rede.

6 - 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://..." )

1- Está criando RDD com todas as linhas de um arquivo do arquivo selecionado. Vai ler os dados do arquivo.

2- Vai montar uma coleção das linhas lidas com seus valores.

3- Faz um mapeamento de chave valor. cada um dos elementos dessa lista ganha um contador iniciado em 1.

4- O comando reduceByKey então irá reduzir todas as ocorrências de palavras repetidas e executar a soma.

5- Vai gravar os dados dentro do hdfs.