



Data Science Academy

www.datascienceacademy.com.br

Data Lake – Design, Projeto e Integração

Computação Distribuída Hadoop MapReduce



MapReduce é a camada de processamento do Hadoop. O modelo de programação MapReduce é projetado para processar grandes volumes de dados em paralelo dividindo o trabalho em um conjunto de tarefas independentes. Você precisa programar a lógica de negócios de acordo com a maneira como o MapReduce funciona e o framework se encarrega do restante. O job (trabalho completo) que é submetido pelo usuário ao mestre, é dividido em pequenas tarefas e atribuído aos escravos.

Os programas MapReduce são escritos em um estilo particular influenciado por construções de programação funcional e expressões específicas para o processamento de listas de dados. No MapReduce, obtemos entradas de uma lista e a convertemos em saída, o que é novamente uma lista. O Hadoop é muito poderoso e eficiente devido ao MapReduce, já que é feito o processamento paralelo, fundamental com grandes conjuntos de dados.

O MapReduce divide o trabalho em pequenas partes, cada uma das quais pode ser feita em paralelo no cluster de servidores. Um problema é dividido em um grande número de problemas menores, cada um dos quais é processado para fornecer saídas individuais. Essas saídas individuais são processadas para fornecer a saída final.

O Hadoop MapReduce é escalonável e também pode ser usado em vários computadores. Muitas máquinas pequenas podem ser usadas para processar tarefas que não puderam ser processadas por uma máquina de maior capacidade.

Vamos estudar o processamento de grandes conjuntos de dados no curso de Machine Learning e IA em Ambientes Distribuídos, mas por que não compreender agora um pouco mais sobre o MapReduce e sua relação com o armazenamento de dados? Então, continue sua leitura.



Terminologias do Apache MapReduce

Vamos compreender diferentes terminologias e conceitos de MapReduce, o que é Mapear e Reduzir, o que é um job, tarefa, tentativa de tarefa, etc.

MapReduce é o componente de processamento de dados do Hadoop. Os programas MapReduce transformam listas de elementos de dados de entrada em listas de elementos de dados de saída. Um programa MapReduce fará isso duas vezes, usando duas atividades de processamento de lista diferentes: Map e Reduce.

Entre Map e Reduce, existe uma pequena fase chamada Shuffle e Sort in MapReduce. Vamos entender as terminologias básicas usadas no MapReduce.

O que é um Job MapReduce?

Um Job MapReduce ou um “programa completo” é uma execução de um Mapeador e Redutor em um conjunto de dados. É uma execução de 2 camadas de processamento, ou seja, mapeador e redutor. Um job MapReduce é um trabalho que o cliente deseja executar. Consiste nos dados de entrada, no Programa MapReduce e nas informações de configuração. Portanto, o cliente precisa enviar dados de entrada, ele precisa escrever o programa MapReduce e definir as informações de configuração (elas foram fornecidas durante a instalação do Hadoop nos arquivos de configuração e também definimos algumas configurações em nosso próprio programa que serão específicas para nosso trabalho de mapeamento e redução).

O que é Tarefa de MapReduce?

Uma tarefa no MapReduce é uma execução de um Mapeador ou Redutor em uma fatia de dados. Também é chamado de Tarefa em andamento (ou TIP - Task-In-Progress). Isso significa que o processamento de dados está em andamento no mapeador ou no redutor.



O que é Task Attempt (Tentativa de Tarefa)?

A Tentativa de Tarefa é uma instância específica de uma tentativa de executar uma tarefa em um nó. Existe a possibilidade de que a qualquer momento qualquer máquina possa cair. Por exemplo, durante o processamento de dados, se algum nó ficar inativo, a estrutura reprogramará a tarefa para algum outro nó. Essa reprogramação da tarefa não pode ser infinita. Há um limite superior para isso também. O valor padrão da tentativa de tarefa é 4. Se uma tarefa (mapeador ou redutor) falhar 4 vezes, a tarefa será considerada como uma tarefa com falha. Para jobs de alta prioridade ou muito grandes, o valor dessa tentativa de tarefa também pode ser aumentado.

Map Abstraction

Vamos entender a forma abstrata do Map em MapReduce, a primeira fase do paradigma MapReduce, o que é um map/mapper, qual é a entrada para o mapeador, como ele processa os dados e o que é gerado pelo mapeador.

O Map usa um par chave/valor como entrada. Independentemente de os dados estarem no formato estruturado ou não estruturado, o framework converte os dados recebidos em chave e valor.

- Chave é uma referência ao valor de entrada.
- Valor é o conjunto de dados no qual operar.

No processamento do Map uma função definida pelo usuário com uma lógica de negócios personalizada de acordo com sua necessidade de processar os dados é aplicada a todos os valores nos dados de entrada.

O Map produz uma nova lista de pares chave/valor, onde uma saída do Map é chamada de saída intermediária e pode ser de tipo diferente do par de entrada. Uma saída do Map é armazenada no disco local de onde é embaralhada (shuffled) e enviada para os nós responsáveis pela operação de Reduce.

Reduce Abstraction

Agora, vamos discutir a segunda fase do MapReduce. A operação Reduce recebe pares de chave/valor intermediários como entrada e processa a saída do mapeador. Normalmente, no redutor, fazemos tarefas de computação como agregação ou soma.

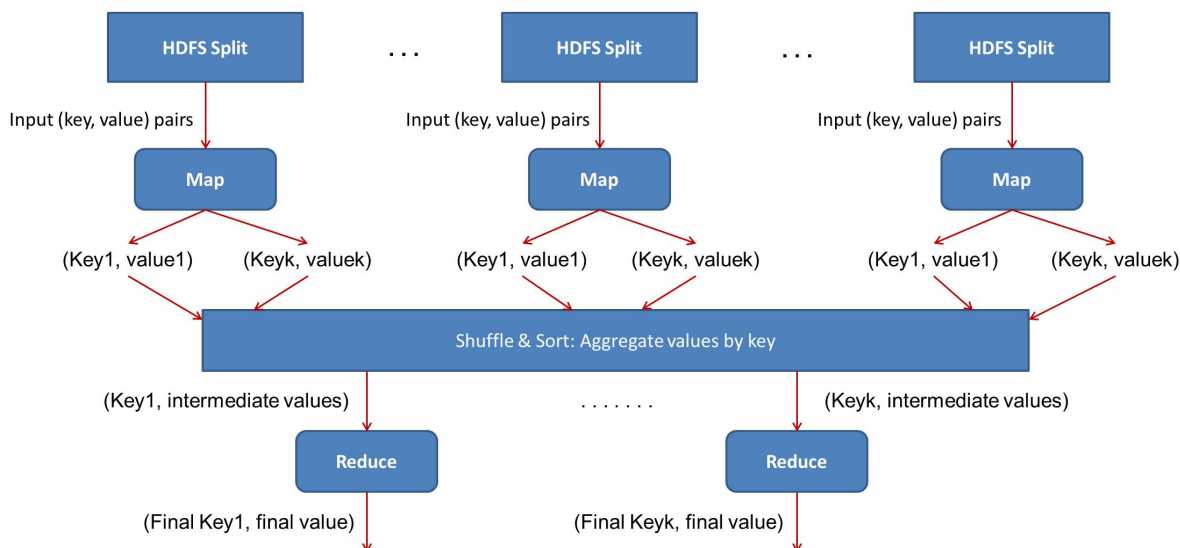
A entrada dada ao redutor é gerada pelo Map (saída intermediária) e os pares chave/valor fornecidos para reduzir são classificados por chave

No processamento da Redução uma função definida pelo usuário com uma lógica de negócios personalizada de acordo com sua necessidade de processar os dados é aplicada para gerar os dados de saída. O iterador fornece os valores de uma determinada chave para a função Reduce.

O Reduce produz uma lista final de pares chave/valor e uma saída de Reduce é chamada de saída Final, podendo ser um tipo diferente do par de entrada. Uma saída do Reduce é armazenada no HDFS.

Como Map e Reduce trabalham juntos?

Vamos entender como o Hadoop Map e o Reduce funcionam juntos. Observe o diagrama abaixo:

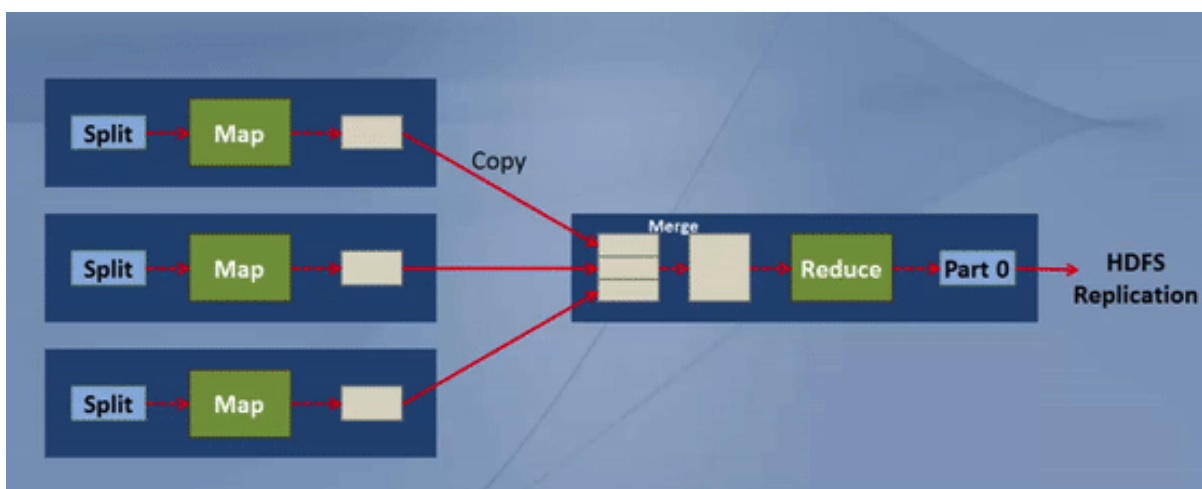


Os dados de entrada fornecidos ao mapeador são processados por meio da função definida pelo usuário gravada no mapeador. Toda a lógica de negócios complexa necessária é implementada no nível do mapeador, de modo que o processamento pesado é feito pelo mapeador em paralelo, pois o número de mapeadores é muito maior que o número de redutores. O mapeador gera uma saída que é um dado intermediário e essa saída serve como entrada para o redutor.

Este resultado intermediário é então processado pela função definida pelo usuário escrita no redutor e a saída final é gerada. Normalmente, no redutor um processamento muito leve é feito. Essa saída final é armazenada no HDFS e a replicação é feita normalmente.

Data Flow

Agora, vamos entender neste fluxo completo de dados do MapReduce, como a entrada é dada ao mapeador, como os mapeadores processam dados, onde os mapeadores gravam os dados, como os dados são “embaralhados” dos nós do mapeador para o redutor, onde os redutores são executados e que tipo de processamento deve ser feito nos redutores.



Como visto no diagrama do fluxo de trabalho do MapReduce no Hadoop, o bloco quadrado é um escravo. Existem 3 escravos na figura. Em todos os 3 os mapeadores serão executados e, em seguida, um redutor será executado em



qualquer 1 nó escravo. Para simplificar a figura, o redutor é mostrado em uma máquina diferente, mas pode ser o mesmo nó do mapeador.

Como Funciona Todo o Processo

Uma entrada para um mapeador é de 1 bloco de cada vez. Uma saída do mapeador é gravada em um disco local da máquina na qual o mapeador está sendo executado. Quando o Map termina, essa saída intermediária viaja para os nós redutores (nó onde o redutor será executado).

O redutor é a segunda fase do processamento em que o usuário pode gravar novamente sua lógica de negócios personalizada. Portanto, uma saída do redutor é a saída final gravada no HDFS.

Por padrão, em um escravo, 2 mapeadores são executados por vez, o que também pode ser aumentado conforme os requisitos. Depende novamente de fatores como hardware do datanode, tamanho de bloco, configuração de máquina, etc. Não devemos aumentar o número de mapeadores além do limite determinado, pois isso diminuirá o desempenho.

O Mapper no Hadoop MapReduce grava a saída no disco local da máquina em que está trabalhando. Esses são os dados temporários. Uma saída do mapeador também é chamada de saída intermediária. Todos os mapeadores estão gravando a saída no disco local. À medida que o Primeiro mapeador termina, os dados (saída do mapeador) estão viajando do nó do mapeador para o nó redutor. Portanto, esse movimento de saída do nó mapeador para o nó redutor é chamado de aleatório.

O redutor também é implantado em qualquer um dos datanodes. Uma saída de todos os mapeadores vai para o redutor. Todas essas saídas de diferentes mapeadores são mescladas para formar uma entrada para o redutor. Esta entrada também está no disco local. Redutor é outro processador onde você pode escrever lógica de negócios personalizada. É o segundo estágio do processamento. Normalmente, para o redutor, escrevemos agregação, somatória e outros tipos de funcionalidades. Por isso, o redutor fornece a saída final que ele grava no HDFS.



Mapear e reduzir são as etapas do processamento. Elas correm um após o outro. Afinal, os mapeadores concluem o processamento e, em seguida, somente o redutor inicia o processamento.

Apesar de 1 bloco estar presente em 3 locais diferentes por padrão, a estrutura permite que apenas 1 mapeador processe 1 bloco. Portanto, apenas um mapeador processará um bloco específico de três réplicas. A saída de cada mapeador vai para todos os redutores no cluster, ou seja, cada redutor recebe entrada de todos os mapeadores. Portanto, a estrutura indica ao redutor que todos os dados foram processados pelo mapeador e agora o redutor pode processar os dados.

Uma saída do mapeador é particionada e filtrada para muitas partições pelo particionador. Cada uma dessas partições vai para um redutor com base em algumas condições. O Hadoop trabalha com o princípio do chave/valor, ou seja, o mapeador e o redutor obtêm a entrada na forma de chave e valor e escrevem a saída também na mesma forma.

Localidade de Dados no MapReduce

Vamos entender o que é a localidade dos dados, como ela otimiza os trabalhos do Map Reduce e como a localização dos dados melhora o desempenho do trabalho.

“Mova a computação para perto dos dados em vez de dados para computação”.

Um cálculo solicitado por um aplicativo é muito mais eficiente se executado próximo aos dados em que opera. Isso é especialmente verdadeiro quando o tamanho dos dados é muito grande. Isso minimiza o congestionamento da rede e aumenta a taxa de transferência do sistema. A suposição é que muitas vezes é melhor mover o cálculo para mais perto de onde os dados estão presentes, em vez de mover os dados para onde o aplicativo está sendo executado. Portanto, o HDFS fornece interfaces para que os aplicativos se aproximem de onde os dados estão presentes.



Como o Hadoop funciona com um grande volume de dados e não é viável mover esse volume pela rede, surgiu o princípio mais inovador de mover os algoritmos para os dados, em vez de dados para os algoritmos. Isso é chamado de localidade de dados.

Ufa! Muita coisa não? Seu trabalho como Engenheiro de Dados é garantir que todo esse processo funcione sem erros! 😊