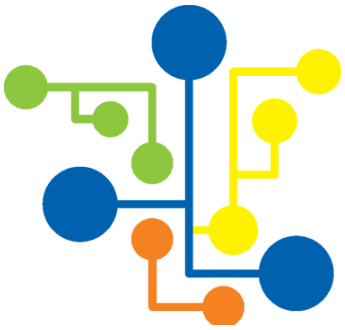


Data Science Academy

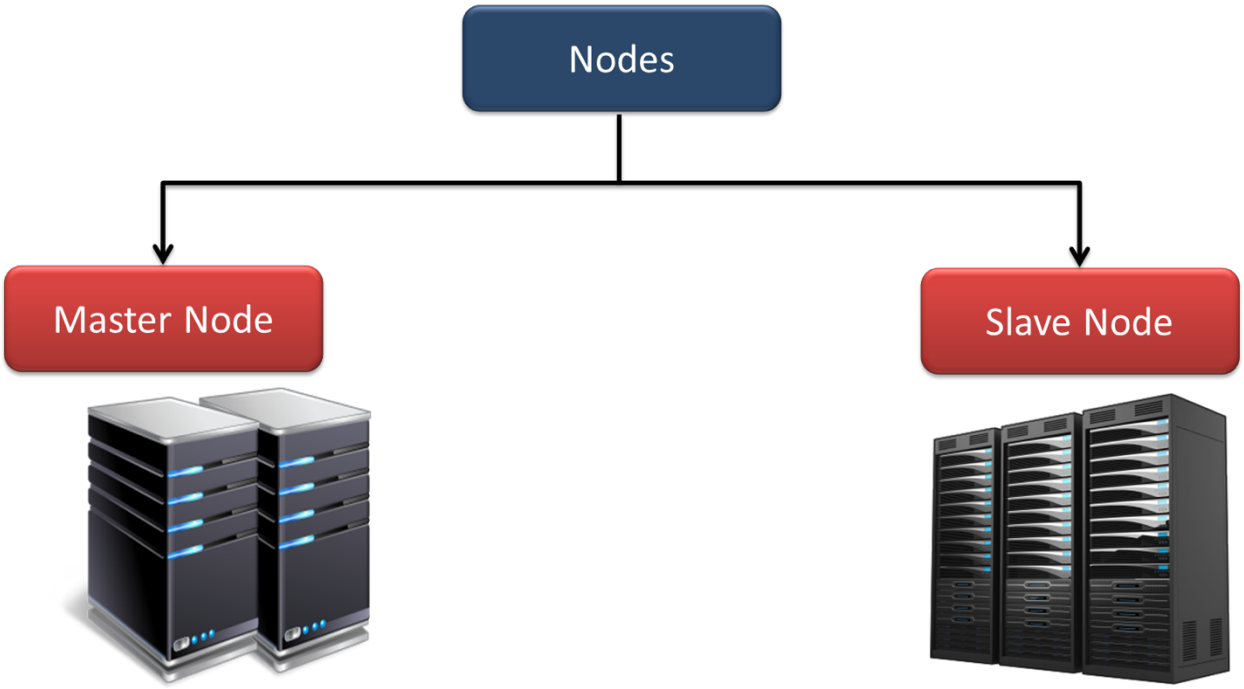
www.datascienceacademy.com.br

Data Lake – Design, Projeto e Integração

Armazenamento Distribuído
Hadoop Distributed File System

O Sistema de Arquivos Distribuído do Hadoop (ou Hadoop Distributed File System – HDFS) é o sistema de armazenamento mais confiável do mundo. O HDFS é o sistema de arquivos do Hadoop projetado para armazenar arquivos muito grandes em um cluster de hardware comum.

Ele foi projetado com o princípio de armazenamento de um menor número de arquivos grandes, em vez de um grande número de arquivos pequenos (como é o caso do sistema de arquivos do seu computador pessoal). O HDFS também fornece uma camada de armazenamento tolerante a falhas para o Hadoop e seus outros componentes e a replicação de dados nos ajuda a atingir esse objetivo. Ele armazena dados de forma confiável, mesmo em caso de falha de hardware e fornece acesso de alto rendimento aos dados, de forma paralela.



Como sabemos, o Hadoop funciona de forma mestre-escravo, com dois tipos de nós. Existem namenode (s) e datanodes no cluster.

Mestre HDFS (Namenode)

O Namenode regula o acesso aos arquivos para os clientes. Ele mantém e gerencia os nós escravos e atribui tarefas a eles. O Namenode executa operações

de namespace do sistema de arquivos, como abrir, fechar e renomear arquivos e diretórios. Deve ser implantado em hardware confiável.

Escravo HDFS (Datanode)

Há vários escravos ou datanodes no Hadoop Distributed File System, que gerenciam o armazenamento de dados. Esses nós escravos são os nós de trabalho reais que executam as tarefas e atendem a solicitações de leitura e gravação dos clientes do sistema de arquivos. Eles também executam a criação, exclusão e replicação de blocos sob instruções do Namenode. Quando um bloco é gravado em um datanode, ele o replica para outro datanode e o processo continua até que o número de réplicas mencionadas seja criado. Datanodes podem ser implantados em hardware commodity e não precisamos implantá-los em hardware muito confiável.

Daemons do HDFS

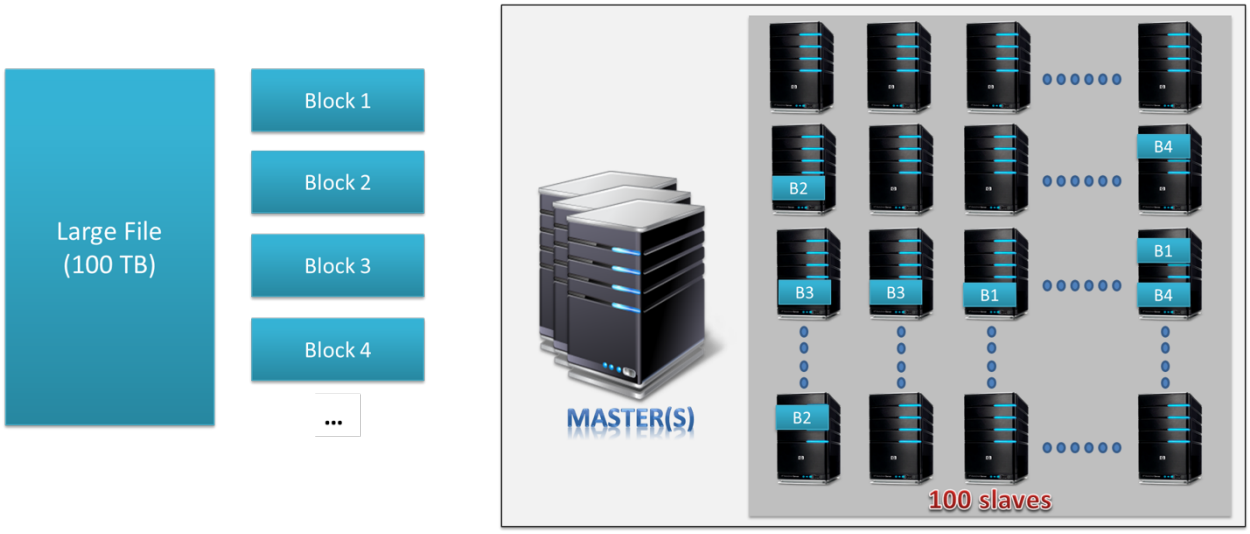
Existem 2 daemons que são executados pelo HDFS para armazenamento de dados:

- Namenode: Este é o daemon que é executado em todos os mestres. O Namenode armazena metadados como nome do arquivo, o número de blocos, número de réplicas, uma localização de blocos, IDs de bloco etc. Esses metadados estão disponíveis na memória no mestre para recuperação mais rápida de dados. No disco local, uma cópia dos metadados está disponível para persistência. Portanto, a memória do Namenode deve ser alta conforme o requisito de arquitetura.

- Datanode: Este é o daemon que é executado no escravo. Esses são nós de trabalho reais que armazenam os dados.

Armazenamento de Dados no HDFS

Sempre que algum arquivo tiver que ser gravado no HDFS, ele será dividido em pequenos pedaços de dados, conhecidos como blocos. O HDFS tem um tamanho de bloco padrão de 128 MB, que pode ser aumentado de acordo com os requisitos. Esses blocos são armazenados no cluster de maneira distribuída em diferentes nós. Isso fornece um mecanismo para o MapReduce processar os dados em paralelo no cluster.



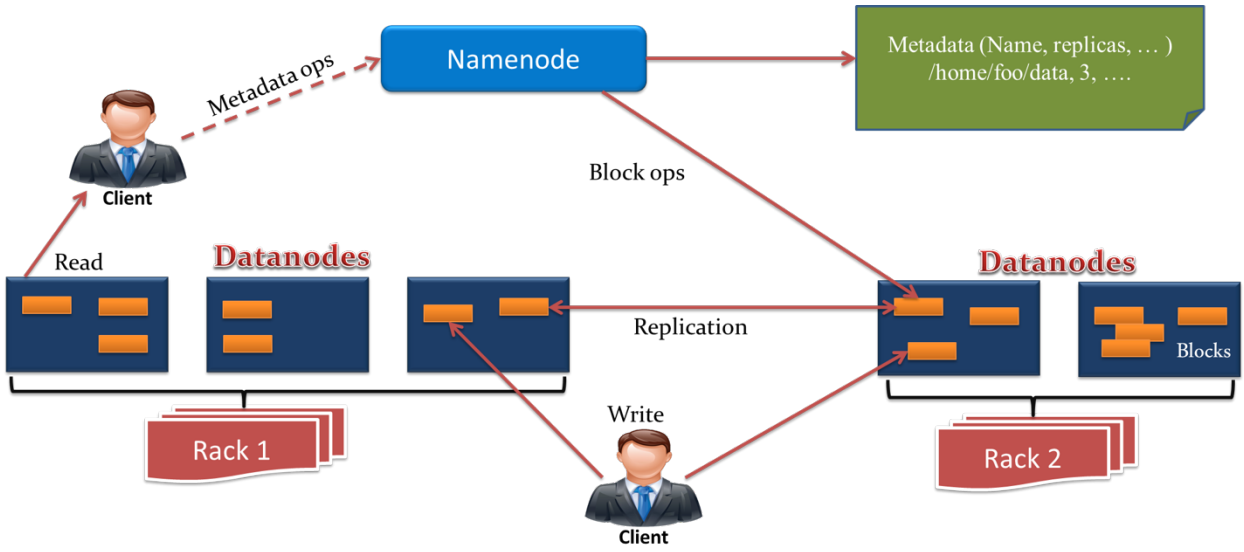
Várias cópias de cada bloco são armazenadas no cluster em nós diferentes. Esta é uma replicação de dados. Por padrão, o fator de replicação HDFS é 3. Ele fornece tolerância a falhas, confiabilidade e alta disponibilidade.

Rack Awareness no HDFS

O Hadoop é executado em um cluster de computadores que geralmente são distribuídos por vários racks. O Namenode coloca réplicas de um bloco em vários racks para melhorar a tolerância a falhas. O Namenode tenta colocar pelo menos uma réplica de um bloco em cada rack, de modo que, se um rack completo ficar inativo, o sistema também permanecerá altamente disponível. A otimização do posicionamento de réplica distingue o HDFS da maioria dos outros sistemas de arquivos distribuídos. O objetivo de uma política de posicionamento de réplica ciente de rack é melhorar a confiabilidade, a disponibilidade e a utilização da largura de banda da rede.

Arquitetura HDFS

A imagem abaixo fornece uma visão completa do Sistema de Arquivos Distribuído do Hadoop. Existe um único Namenode que armazena metadados e existem vários datanodes que fazem o trabalho de armazenamento real. Os nós são organizados em racks e as réplicas de blocos de dados são armazenadas em diferentes racks no cluster para fornecer tolerância a falhas. Para ler ou gravar um arquivo no HDFS, o cliente precisa interagir com o Namenode. Os aplicativos HDFS precisam de um modelo de acesso write-once-read-many para arquivos. Um arquivo criado e gravado não pode ser editado.



O Namenode armazena metadados e os datanodes armazenam dados reais. O cliente interage com o Namenode para que qualquer tarefa seja executada, pois o Namenode é a peça central do cluster.

Existem vários datanodes no cluster que armazenam dados do HDFS no disco local. O datanode envia uma mensagem de *heartbeat* para o Namenode periodicamente para indicar que ele está ativo. Além disso, ele replica dados para outro datanode conforme o fator de replicação.

Blocos

O HDFS divide arquivos enormes em pequenos pedaços conhecidos como blocos. Um bloco é a menor unidade de dados em um sistema de arquivos. Nós (cliente e admin) não temos nenhum controle sobre o bloco ou sobre a localização do bloco. O Namenode toma todas as decisões.

O tamanho de bloco padrão do HDFS é de 128 MB, o que pode ser aumentado conforme o requisito. Isso é diferente do sistema de arquivos do SO, no qual o tamanho do bloco é de 4 KB.

Se o tamanho dos dados for menor que o tamanho do bloco do HDFS, o tamanho do bloco será igual ao tamanho dos dados. Por exemplo, se o tamanho do arquivo for 129 MB, serão criados 2 blocos para ele. Um bloco será de tamanho padrão de 128 MB e o outro será de apenas 1 MB e não 128 MB, pois irá desperdiçar o espaço (aqui o tamanho do bloco é igual ao tamanho dos dados). O Hadoop é inteligente o suficiente para não desperdiçar 127 MB. Por isso, está alocando um bloco de 1 MB somente para dados de 1 MB.

A principal vantagem de armazenar dados em tal tamanho de bloco é que ele economiza tempo de busca de disco e outra vantagem é no caso de processamento map-reduce que processa 1 bloco de cada vez. Portanto, um processo map-reduce processa dados grandes por vez.

No HDFS, o arquivo é dividido em blocos e cada bloco é armazenado em diferentes nós com 3 réplicas padrão de cada bloco. Cada réplica de um bloco é armazenada em nó diferente para fornecer um recurso tolerante a falhas e o posicionamento desses blocos no nó diferente é decidido pelo Namenode. O Namenode faz com que seja distribuído o máximo possível. Ao colocar um bloco em um datanode, ele considera quanto um determinado datanode está carregado nesse momento.

Replicação

O HDFS cria cópias de cada bloco. Isso é chamado de replicação. Todos os blocos são replicados e armazenados em nós diferentes no cluster. Ele tenta colocar pelo menos uma réplica em cada rack. O que você quer dizer com rack?

Datanodes são organizados em racks. Todos os nós em um rack são conectados por um único switch, portanto, se um switch ou rack completo estiver inoperante, os dados poderão ser acessados de outro rack.

Como visto anteriormente, o fator de replicação padrão é 3 e isso pode ser alterado para os valores necessários de acordo com os requisitos da arquitetura editando os arquivos de configuração (hdfs-site.xml)

Acesso de Alto Rendimento aos Dados

O HDFS fornece acesso de alto rendimento aos dados. Throughput é a quantidade de trabalho realizado em um tempo unitário. Ele descreve a rapidez com que os dados estão sendo acessados no sistema e é geralmente usado para medir o desempenho do sistema. Quando queremos realizar uma tarefa ou uma ação, o trabalho é dividido e compartilhado entre diferentes sistemas. Assim, todos os sistemas estarão executando as tarefas atribuídas a eles de forma independente e paralela. Então, o trabalho será concluído em um período muito curto de tempo. Desta forma, o HDFS oferece um bom rendimento (bom Throughput). Ao ler os dados em paralelo, diminuímos o tempo real para ler dados tremendamente.

Ufa! Muita informação? Então relaxe, pois tem mais pela frente! ☺