



Data Science Academy

www.datascienceacademy.com.br

Data Lake – Design, Projeto e Integração

Operações de Leitura e Escrita no HDFS



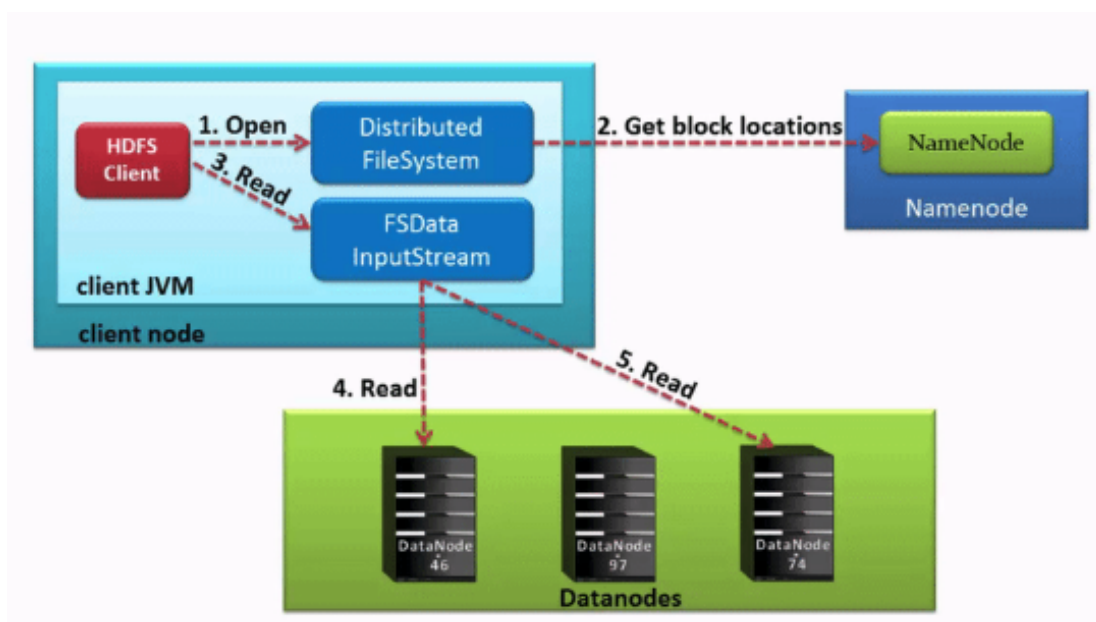
No Hadoop, precisamos interagir com o sistema de arquivos por programação ou por interface de linha de comando (CLI).

Mas há uma boa notícia. O HDFS tem muitas semelhanças com o sistema de arquivos Linux. Assim, podemos fazer quase todas as operações no HDFS do mesmo modo que podemos fazer com um sistema de arquivos local, como criar um diretório, copiar o arquivo, alterar permissões, etc., e usando quase os mesmos comandos (você fez o curso de Linux, bônus deste curso de Data Lake, certo?)

Ele também fornece diferentes direitos de acesso, como ler, escrever e executar para usuários e grupos. Vejamos as duas operações, leitura e escrita em mais detalhes.

Operação de Leitura (Read)

Sempre que um cliente deseja ler qualquer arquivo do HDFS, o cliente precisa interagir com o Namenode, já que o Namenode é o único local que armazena metadados sobre nós de dados. O Namenode especifica o endereço ou a localização dos escravos onde os dados são armazenados. O cliente irá interagir com os nós de dados especificados e ler os dados de lá. Para fins de segurança/autenticação, o Namenode fornece o token ao cliente, que mostra ao nó de dados para ler o arquivo. Observe o diagrama abaixo com esse processo detalhado:



Na operação de leitura, se o cliente quiser ler os dados armazenados no HDFS, ele precisará interagir primeiro com o Namenode. Então, o cliente interage com a API do sistema de arquivos distribuídos e envia uma solicitação para o Namenode enviar o local do bloco. O Namenode verifica se o cliente tem privilégios suficientes para acessar os dados ou não e compartilha o endereço no qual os dados são armazenados no datanode.

Com o endereço, o Namenode também compartilha um token de segurança com o cliente, que ele precisa mostrar para o datanode antes de acessar os dados para fins de autenticação (estudaremos isso em mais detalhes no curso Segurança e Alta Disponibilidade de Dados, da Formação Engenheiro de Dados).



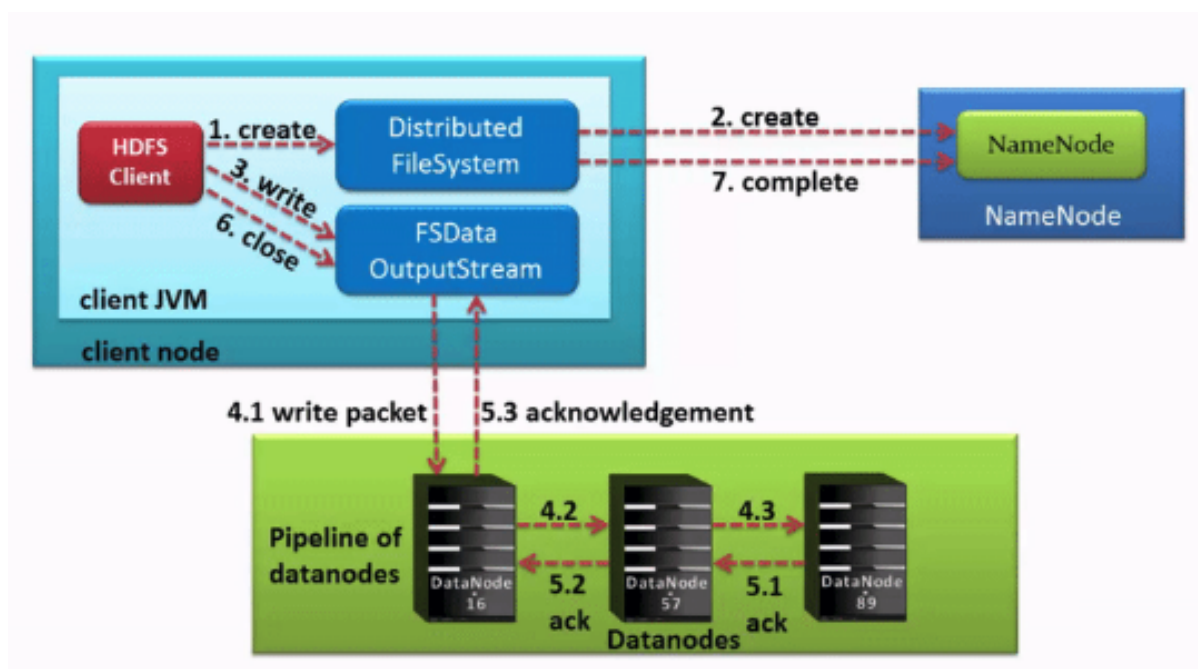
Quando um cliente acessa o datanode para ler o arquivo, depois de verificar o token, o datanode permite que o cliente leia esse bloco em particular. Um cliente então abre o fluxo de entrada (Input Stream) e começa a ler os dados dos datanodes especificados. Portanto, dessa maneira, o cliente lê dados diretamente do datanode.

Se durante a leitura de um arquivo do datanode, o servidor cair repentinamente, então o cliente irá novamente para o Namenode, que compartilhará outro local onde esse bloco esteja presente.

Operação de Escrita (Write)

Como visto durante a leitura de um arquivo, o cliente precisa interagir com o Namenode. Da mesma forma, para escrever um arquivo, o cliente também precisa interagir com o Namenode, que fornece o endereço dos escravos em que os dados devem ser escritos pelo cliente.

Quando o cliente termina de gravar o bloco, o escravo começa a replicar o bloco em outro escravo, que copia o bloco para o terceiro escravo. Esse é o caso quando o fator de replicação padrão de 3 é usado. Depois que a replicação necessária é criada, é enviada uma confirmação final para o cliente. Portanto, o processo de autenticação é semelhante ao visto na operação de leitura.



Sempre que um cliente precisa gravar algum dado, ele precisa interagir com o Namenode. Assim, o cliente interage com a API do sistema de arquivos distribuído e envia uma solicitação para o Namenode enviar o local do escravo.

O Namenode compartilha o local em que os dados devem ser gravados. Em seguida, o cliente interage com o datanode no qual os dados serão gravados e começa a gravar os dados por meio do fluxo de saída de dados (Output Stream). Depois que os dados são gravados e replicados, o datanode envia uma



confirmação para o cliente informando que os dados estão gravados completamente.

Assim que o cliente terminar de escrever o primeiro bloco, o primeiro datanode copiará o mesmo bloco para outro datanode. Assim, este datanode depois de receber o bloco, começa a copiar este bloco para o terceiro datanode. O terceiro datanode então envia uma confirmação para segundo, o segundo datanode envia uma confirmação para o primeiro datanode e, em seguida, o primeiro datanode envia a confirmação final (no caso do fator de replicação padrão).

O cliente está enviando apenas 1 cópia de dados, independentemente do nosso fator de replicação, enquanto os datanodes replicam os blocos. Portanto, a gravação do arquivo no HDFS não é dispendiosa, pois vários blocos estão sendo gravados em vários datanodes.