DATA LAKE

DESIGN, PROJETO E INTEGRAÇÃO

Data Lake Armazenamento de Dados

Data Lake
Armazenamento de Dados

Data Lake
Armazenamento de Dados

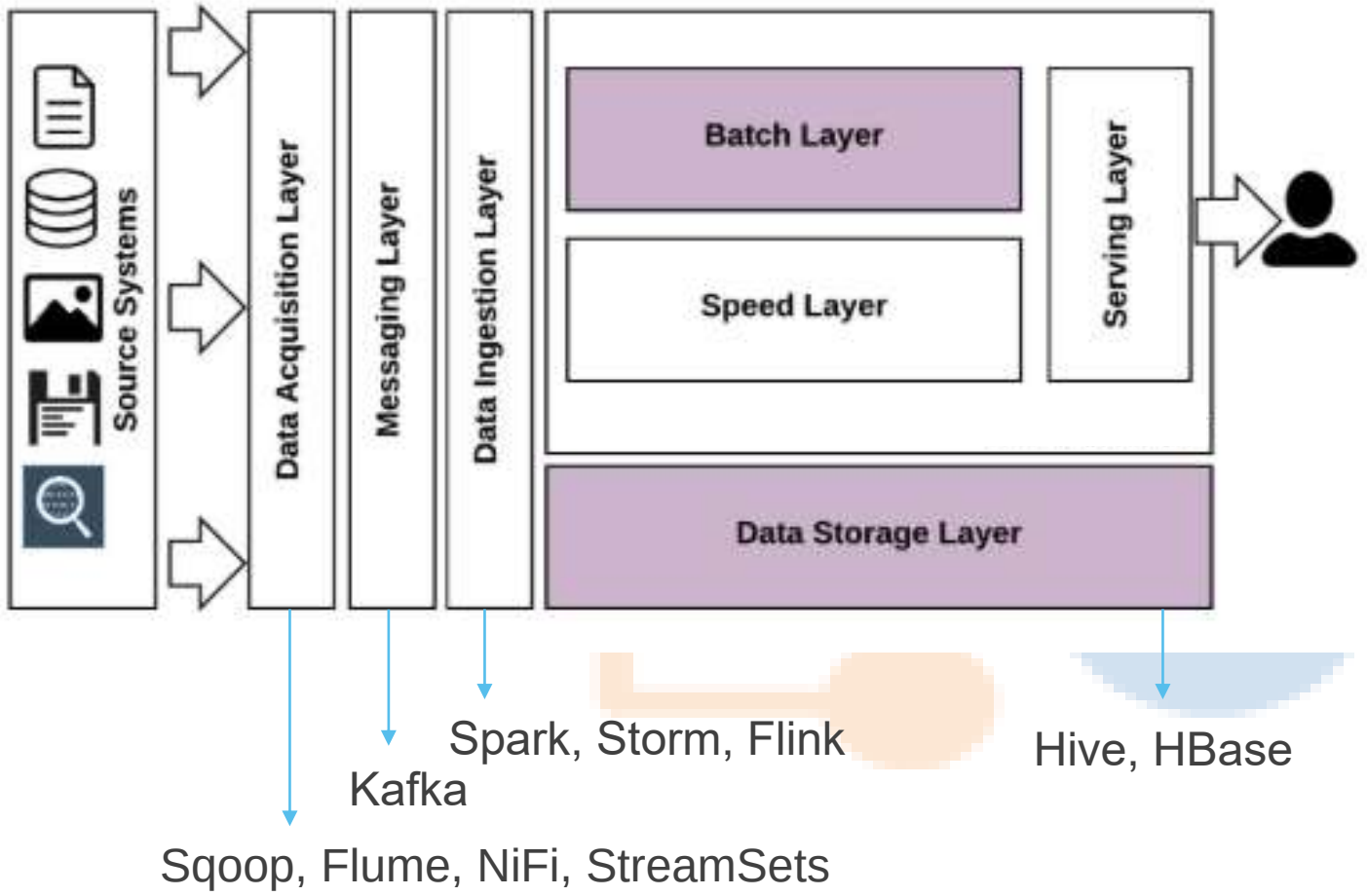
Armazenar os dados em estado bruto

Limpar e organizar os dados antes do armazenamento

Contexto no Data Lake Armazenamento de Dados

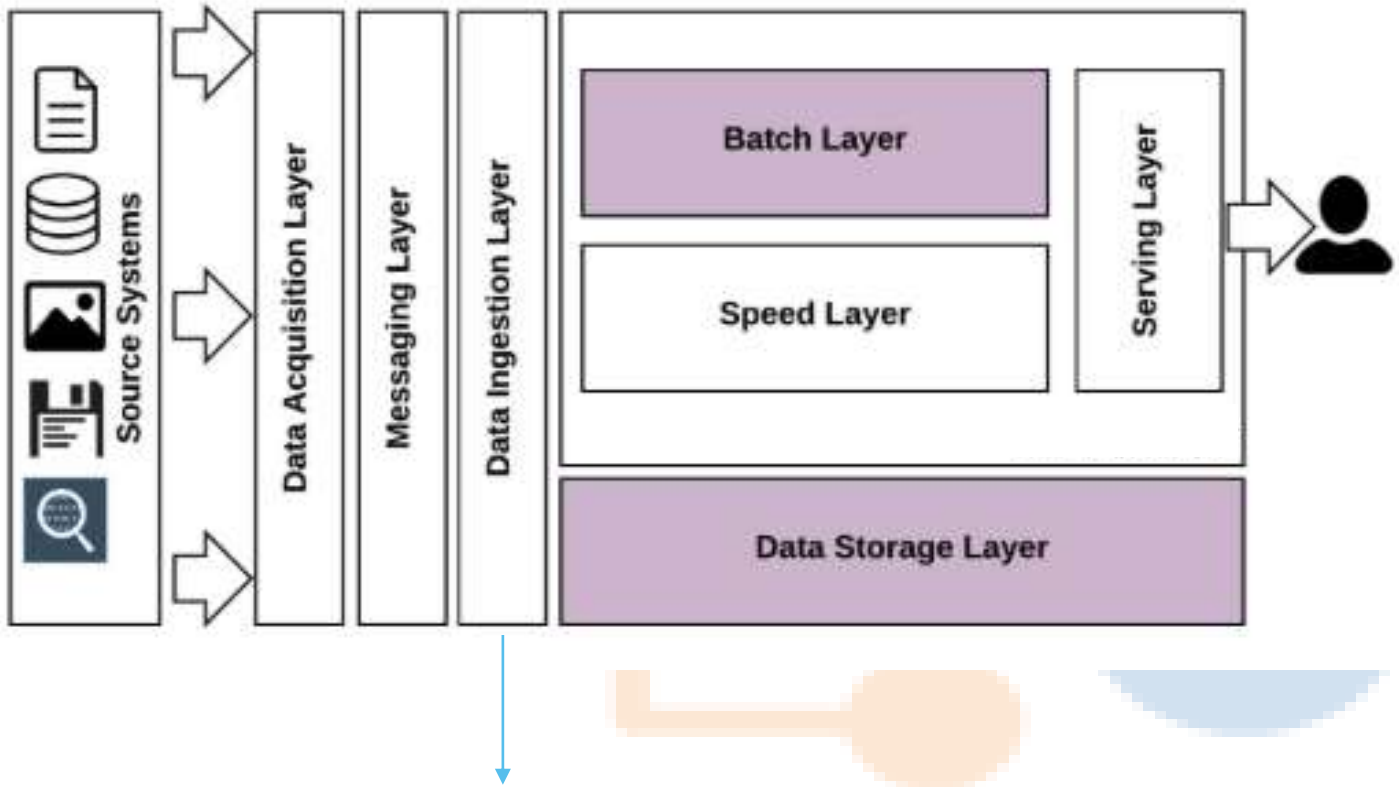
Contexto no Data Lake

Armazenamento de Dados



Contexto no Data Lake

Armazenamento de Dados



Obs: A camada de ingestão de dados, que engloba o processamento dos dados em batch ou em tempo real, será estudada em detalhes no curso Machine Learning e IA em Ambientes Distribuídos, curso número 4 da Formação Engenheiro de Dados!

Objetivo da Camada de Armazenamento Persistência dos Dados

Persistência dos Dados

A camada de armazenamento é responsável pela persistência dos dados e deve estar pronta para escalabilidade sempre que necessário.

Persistência dos Dados

Usamos na camada de armazenamento de dados, qualquer tecnologia que permita a persistência de dados:

Persistência dos Dados

IOPS (Input/Output Operations Per Second)

SSD (Solid State Drive)

HDD (Hard Disk Drive)

SAN (Storage Area Network)

O Que é o Apache Hadoop?

O Que é o Apache Hadoop?

Apache Hadoop é um framework capaz de usar um cluster de computadores de baixo custo, para computação distribuída.

O armazenamento distribuído é feito com o HDFS, enquanto o processamento distribuído pode ser feito com o MapReduce (ou algum outro framework como o Apache Spark).

O Que é o Apache Hadoop?

Apache Hadoop pode ser uma das melhores opções para a camada de armazenamento de dados no Data Lake.

Por que Usar o Hadoop na Camada de Armazenamento?

Por que Usar o Apache Hadoop na Camada de Armazenamento?

- O Hadoop foi concebido para tratar grandes volumes de dados, estruturados, semi-estruturados e não estruturados.
- Em geral, tem um custo menor de implementação, pode ser usado com máquinas de baixo custo.
- A Comunidade Apache oferece amplo suporte ao Hadoop.
- O Hadoop pode ser segmentado. Podemos usar o HDFS na camada de armazenamento e Pig, Hive e Spark na camada de processamento e análise de dados, por exemplo.
- O Hadoop foi pensado para ser tolerante a falhas.
- O Hadoop trabalha com o conceito de “schemaless”, ou seja, não precisamos organizar os dados antes do armazenamento.
- O Hadoop pode ser implementado on-premises ou em nuvem.

Apache Hive e Apache HBase na Camada de Armazenamento

Apache Hive e Apache HBase na Camada de Armazenamento

Os dados estão armazenados no HDFS.

E agora?

Podemos usar Hive e HBase para manipular os dados no HDFS.

Apache Hive
Open Source Data Warehouse

Apache Hive

Open Source Data Warehouse

Apache Hive
Open Source Data Warehouse

O Apache Hive é um sistema de Data Warehouse de código aberto criado sobre o Hadoop para consultar e analisar grandes conjuntos de dados armazenados em arquivos no HDFS.

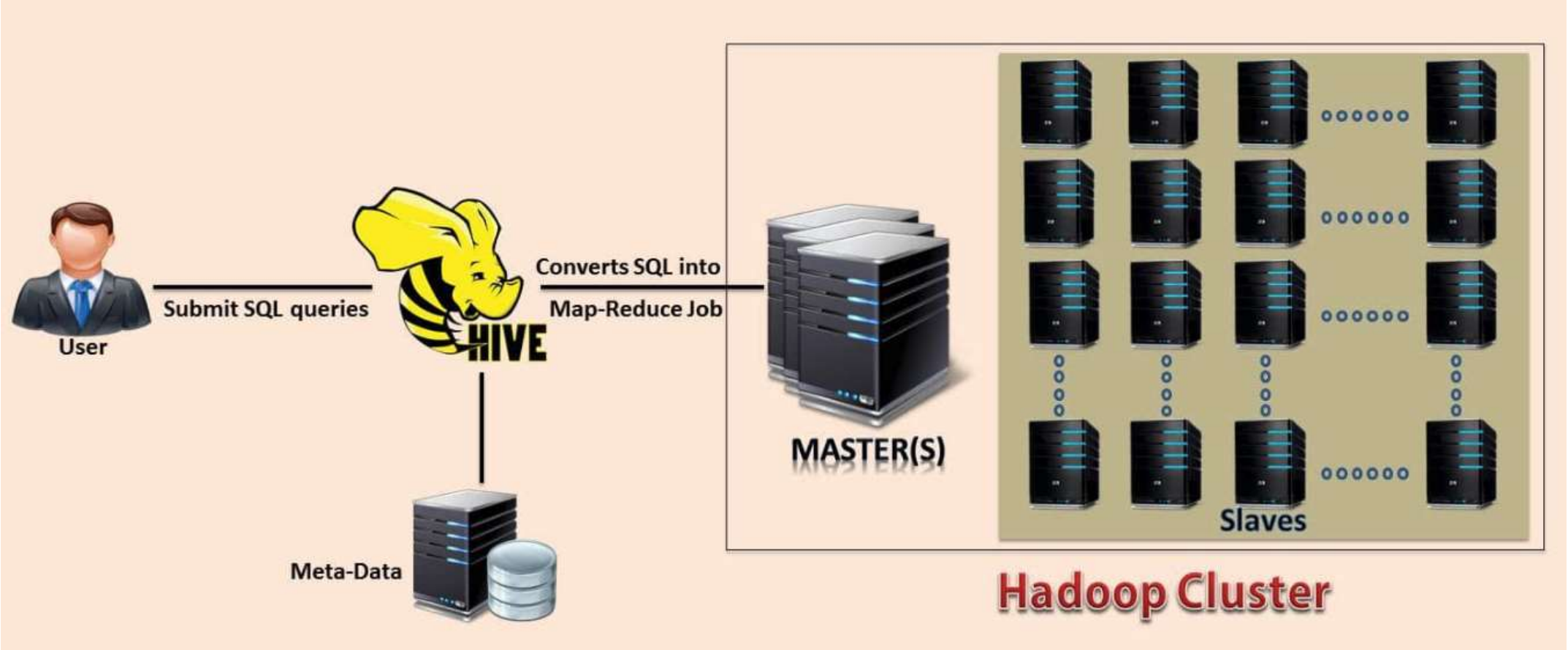
O Hive processa dados estruturados e semiestruturados no Hadoop.

Apache Hive
Open Source Data Warehouse

- Desenvolvido pelo Facebook
- Flexibilidade e evolução de schema
- Tabelas podem ser divididas em partes e balanceadas
- Tabelas do Apache Hive são definidas diretamente no HDFS
- Drivers JDBC / ODBC estão disponíveis

Arquitetura do Apache Hive

Arquitetura do Apache Hive



Arquitetura do Apache Hive

Driver

Age como um controlador que recebe as instruções do HiveQL. O driver inicia a execução da instrução criando sessões.

Metastore

Armazena metadados para cada uma das tabelas, como seu esquema e localização.

Compiler

Executa a compilação da consulta do HiveQL. Isso converte a consulta em um plano de execução.

Optimizer

Realiza várias transformações no plano de execução para fornecer o DAG otimizado.

Arquitetura do Apache Hive

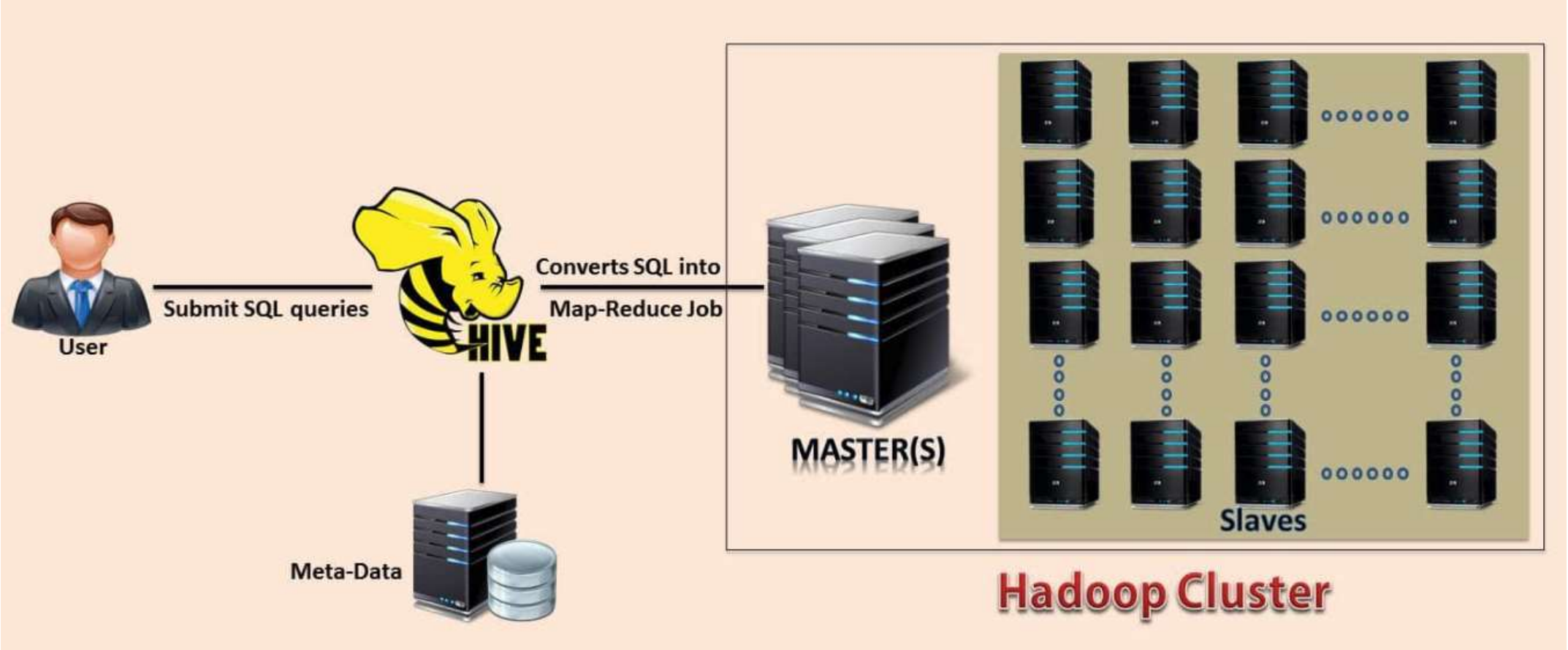
Executor

Depois de concluída a compilação e a otimização, o executor executa as tarefas. Executor cuida do pipelining das tarefas.

CLI, UI e Thrift Server

A CLI (interface da linha de comandos) fornece uma interface de usuário para um usuário externo interagir com o Hive. O servidor Thrift no Hive permite que os clientes externos interajam com o Hive em uma rede, semelhante aos protocolos JDBC ou ODBC.

Arquitetura do Apache Hive



Apache Hive
Shell e Principais Características

Apache Hive
Shell e Principais Características

Hive em Non-Interactive Mode

Hive Shell pode ser executado no modo não interativo, com opção -f podemos especificar a localização de um arquivo que contém consultas HQL. Por exemplo, hive -f my-script.q

Hive em Interactive Mode

O Hive Shell também pode ser executado no modo interativo. Nesse modo, precisamos ir diretamente para o shell do hive e executar as consultas nele. No shell do Hive, podemos enviar as consultas necessárias manualmente e obter o resultado.

Apache Hive
Shell e Principais Características

Principais Características do Hive:

- O Hive fornece resumo, consulta e análise de dados de maneira muito mais fácil.
- O Hive suporta tabelas externas que tornam possível processar dados sem realmente armazenar no HDFS.
- O Apache Hive se encaixa perfeitamente no requisito de interface de baixo nível do Hadoop.
- Ele também suporta o particionamento de dados no nível de tabelas para melhorar o desempenho.
- O Hive tem um otimizador baseado em regras para otimizar os planos lógicos.
- É escalável e extensível.
- O uso do HiveQL não requer nenhum conhecimento da linguagem de programação. O conhecimento da consulta SQL básica é suficiente.
- Podemos processar facilmente dados estruturados no Hadoop usando o Hive.
- Consultar no Hive é muito simples, pois é semelhante ao SQL.
- Também podemos executar consultas Ad-hoc para a análise de dados usando o Hive.

Apache HBase NoSQL Database

Apache HBase - NoSQL Database

Apache HBase - NoSQL Database

Apache HBase é um banco de dados distribuído orientado por coluna não relacional que é executado sobre o HDFS. É um banco de dados de código aberto NoSQL que armazena dados em linhas e colunas.

Uma célula é a interseção de linhas e colunas.

Apache HBase - NoSQL Database

Para acompanhar as alterações na célula, o controle de versão possibilita a recuperação de qualquer versão do conteúdo. O controle de versão faz diferença entre as tabelas HBase e o RDBMS (Relational DataBase Management System).

Cada valor de célula inclui um atributo “version”, que nada mais é do que um timestamp que identifica a célula.

Apache HBase - NoSQL Database

Por que usar o HBase?

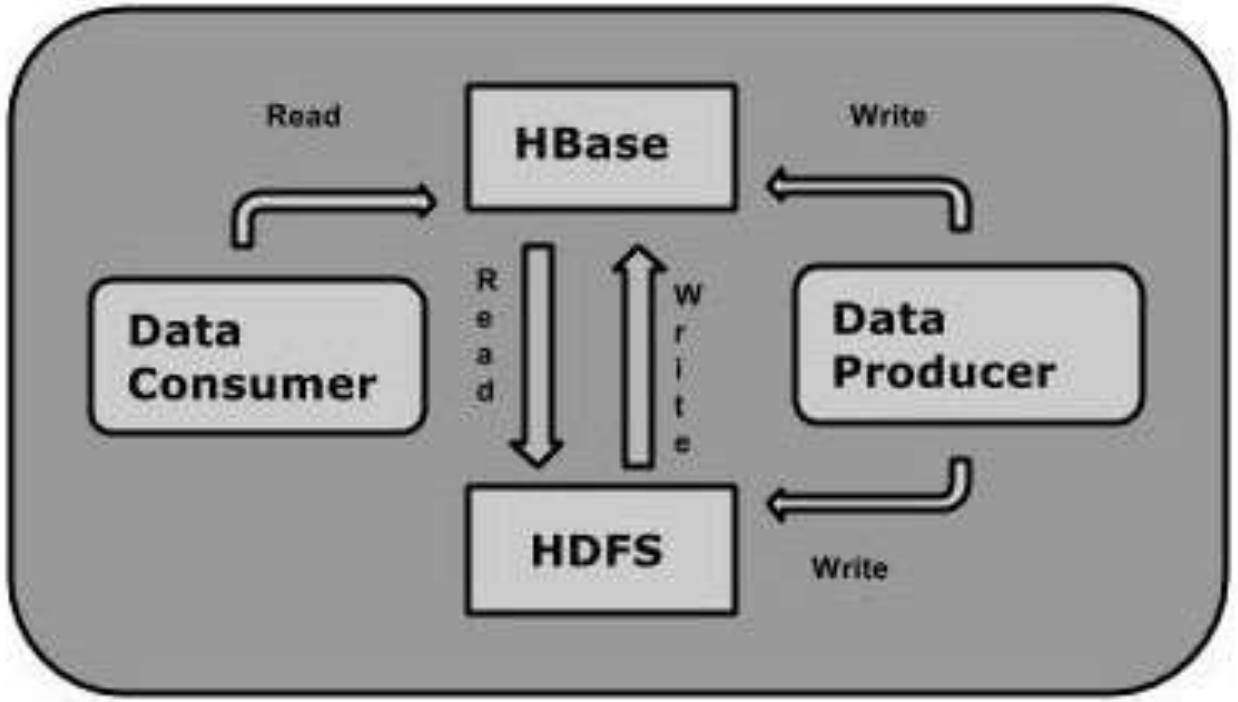
De acordo com o analista do Gartner, Merv Adrian,

“Qualquer um que queira manter os dados dentro de um ambiente HDFS e quiser fazer algo além da leitura de força bruta de todo o sistema de arquivos [com MapReduce] precisa experimentar o HBase. Para acesso aleatório, você precisa ter o HBase.”

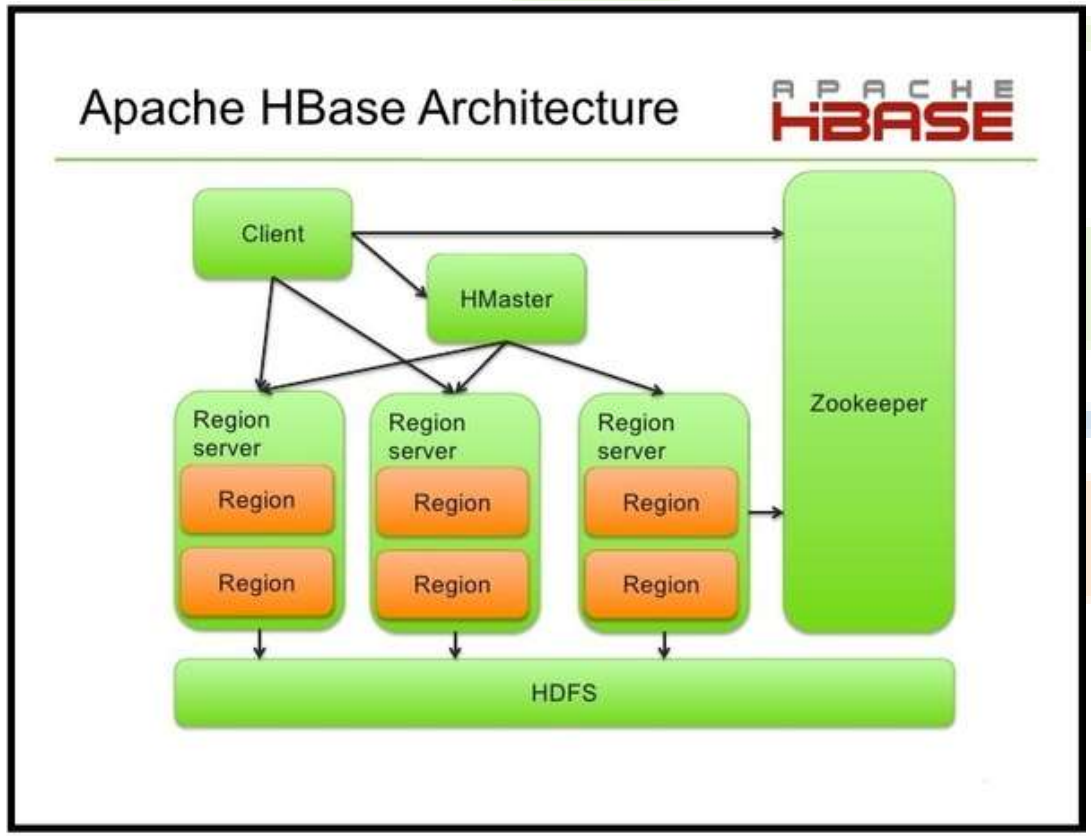
Ele permite leituras e gravações aleatórias rápidas que não podem ser manipuladas pelo Hadoop.

Apache HBase Arquitetura

Apache HBase - Arquitetura



Apache HBase - Arquitetura



- O HMaster lida com o balanceamento de carga em todos os Region Servers e mantém o estado do cluster do Hadoop. Não faz parte do caminho real de armazenamento ou recuperação de dados.
- Os Region Servers são nós que são implementados em cada máquina e hospedam solicitações de E/S de dados e processos.
- O ZooKeeper é um servidor de código aberto que permite uma coordenação distribuída confiável. É um serviço centralizado que mantém informações de configuração, fornecendo sincronização distribuída e fornecendo serviços de grupo.

Muito Obrigado.

É um prazer ter você aqui.
Tenha uma excelente jornada de aprendizagem.