



ENG DE DADOS COM HADOOP E SPARK 3



1. Planejando e Configurando um Cluster Hadoop

2. Usando MapReduce em Grandes Volumes de Dados

3. Armazenamento de dados com HBase e Hive

4. Conectividade ETL com o Sistema Hadoop

5. Administração e Manutenção do Hadoop

6. Hadoop Machine Learning com Apache Mahout

7. Apache Hadoop e Apache Spark

3. ARMAZENAMENTO DE DADOS COM HBASE E HIVE



APACHE HBASE

Banco de dados NoSQL
(Columns Data Model)



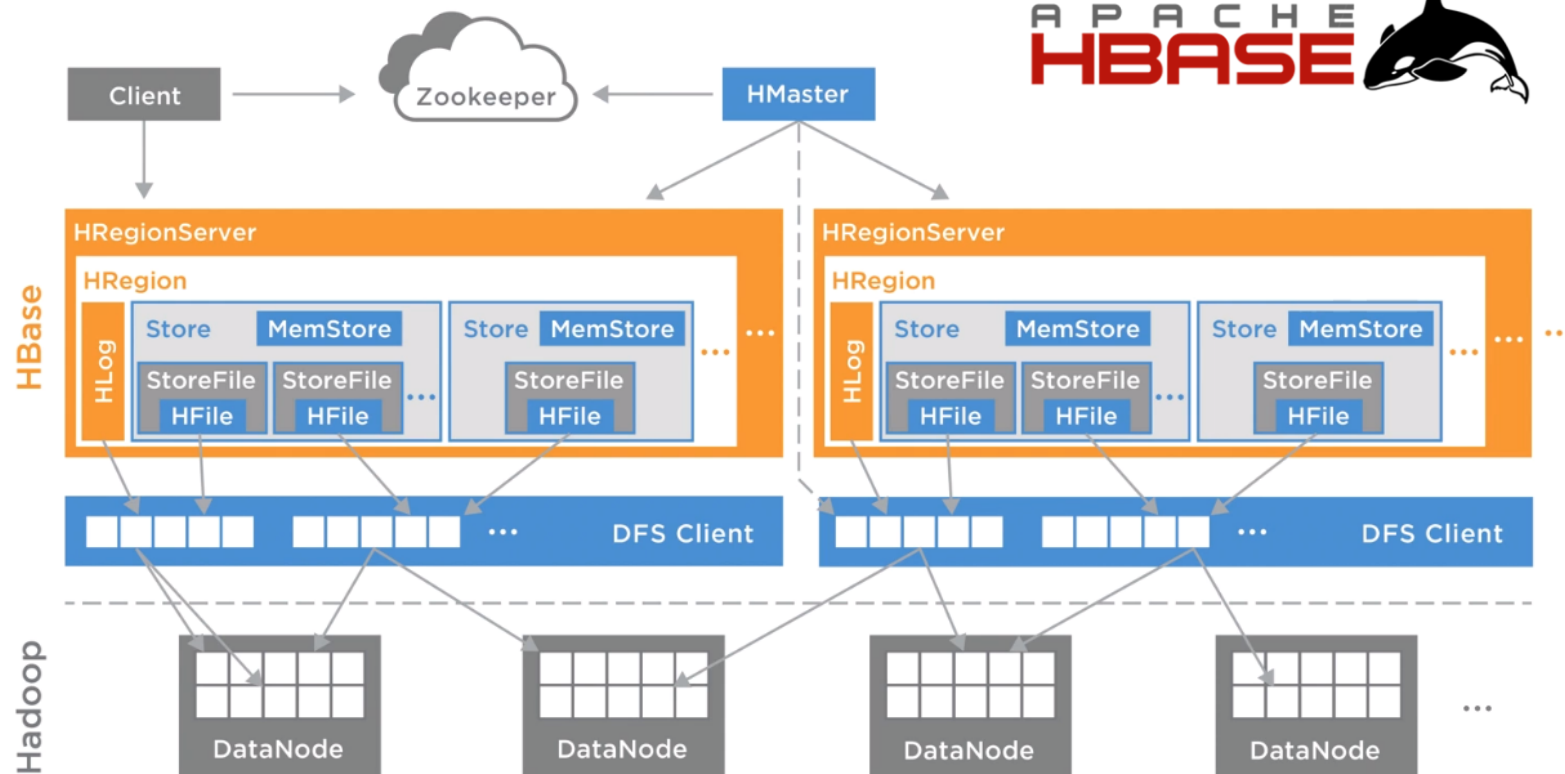
NoSQL

APACHE
HBASE

mongoDB



redis



Row Key	Students		Branch	
StudentID	Name	Age	Bname	GPA
100	Ram	18	CSE	7.9
101	Sham	17	ECE	8
102	John	18	EEE	7.5
103	Sam	17	CSE	8.5

Row Key

Column Families

Column

Cells

Representa os dados
de uma coluna!

CONHECENDO O APACHE HBASE



- O HBase é um banco de dados escalável, distribuído, open source, e que é executado sobre o Hadoop HDFS.
- O Hbase é um banco de dados NoSQL orientado a coluna, não quer um **Schema** de dados pré definido.
- Podemos imaginar o Hbase como uma planilha multidimensional, que pode receber dados com qualquer tipo de estrutura.
- Suporta recuperação automática à falhas, escalabilidade, particionamento automático e esquema de dados totalmente flexíveis.
- **Hbase é mais um “DataStore” do que um “DataBase”.** Possui lookUp e update permitindo acesso veloz à grandes tabelas, com conjuntos de dados.
- **Hbase é um BD distribuído, open-source, não relacional, inspirado no Google Big Table.**

O Hbase não
recomenda a
criação de mais de
4 tabelas!

Poucas
colunas e
muitas linhas!

CARACTERÍSTICAS DO APACHE HBASE



- O Hbase possui **Escalabilidade Horizontal** – aumento do número de máquinas que rodam em paralelo.
- NOTE: Escalabilidade vertical – aumento de recursos de um único computador. (Ex: de 4GB para 8GB) (existe um limite).
- Processos consistentes de leitura/escrita (Hbase Read/Hbase Write).
- Particionamento automático (para que as queries sejam mais rápidas).
- Recuperação automática de falhas.
- Java API para acesso aos dados (via **JDBC**)

HBASE X HDFS



HBase	HDFS
HBase é um banco de dados NoSQL construído para trabalhar sobre o HDFS.	Sistema de arquivos distribuído para armazenamento de grandes conjuntos de dados.
Suporta consultas a grandes tabelas de dados.	Não suporta consultas a registros individuais de dados.
Baixa latência de acesso aos dados, mesmo em tabelas de bilhões de registros.	Alta latência e processamento em batch.
HBase armazena dados em formato key/value.	Armazena os dados em arquivos.

- ↓
- Ferramenta de consulta!
 - Acesso mais específico aos dados.

↘

Armazenamento, não uma ferramenta de consulta!

HBASE X BANCO DE DADOS RELACIONAL



HBase	RDBMS
Utiliza regiões.	Utiliza tabelas.
Suporta o filesystem HDFS.	Suporta filesystems FAT, NTFS, EXT, NFS.
Conceito de Write-Ahead Logs (WAL) para armazenar alterações nos dados.	Conceito de commit logs para armazenar as alterações nos dados.
A coordenação dos processos é feita pelo Apache Zookeeper.	A coordenação dos processos é feita pelo sistema gerenciador de bancos de dados (Oracle, SQL Server, MySQL, etc...)
Linhas são identificadas unicamente pelas rowkeys .	Linhas são identificadas unicamente por chaves primárias.
Regiões podem ser particionadas.	Tabelas podem ser particionadas.
Conceito de linha, família de colunas , coluna e célula.	Conceito de linha, coluna e célula.
Suporta bilhões de registros.	Apresenta problemas de performance com bilhões de registros.

QUANDO USAR E QUANDO NÃO USAR O HBASE?



Quando utilizar o Hbase?

Dados não-estruturados ou semi-estruturados

Alta escalabilidade

Dados versionados

Quando é necessário acesso baseado em chave

Alto volume de dados devem ser armazenados

Armazenamento de dados orientado a coluna

Quando **NÃO** utilizar o Hbase?

Poucas linhas devem ser armazenadas

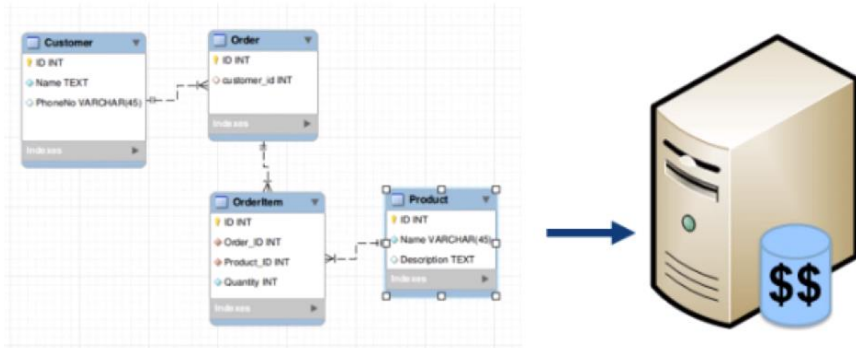
Não for necessário realizar consultas cruzadas (SQL Joins)

Cluster com poucas máquinas

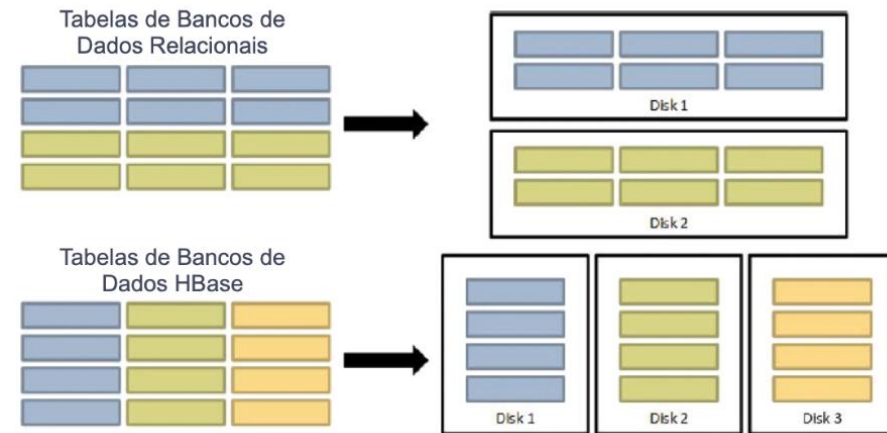
MODELO DE DADOS DO HBASE



RDBMS - Scale UP approach



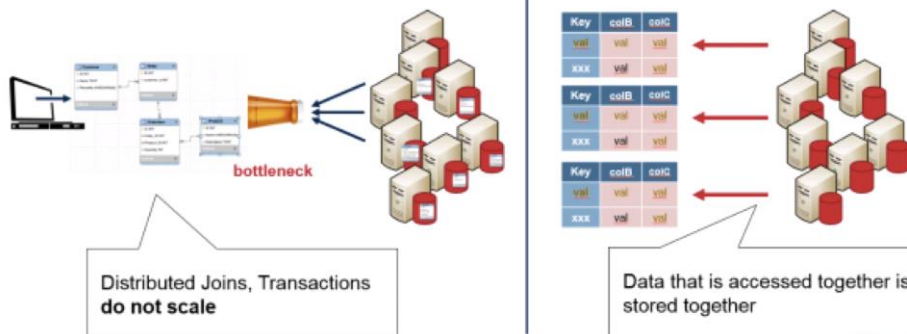
Escalabilidade Vertical



RDBMS

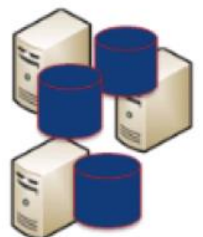
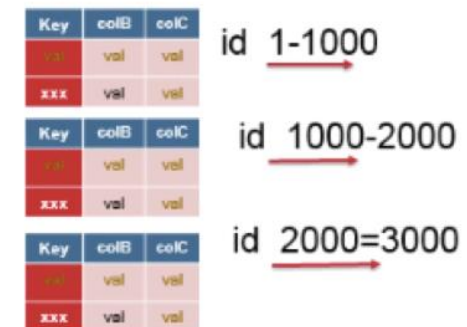
Storage Model

HBase



Tabelas divididas em
partições e distribuídas
no cluster

Processamento de Big Data!



Escalabilidade Horizontal

FAMILIA DE COLUNAS



Row Key	Customer		Sales	
Customer Id	Name	City	Product	Amount
101	John White	Los Angeles, CA	Chairs	\$400.00
102	Jane Brown	Atlanta, GA	Lamps	\$200.00
103	Bill Green	Pittsburgh, PA	Desk	\$500.00
104	Jack Black	St. Louis, MO	Bed	\$1600.00

Célula

Hfile

Column Families

Identificação da célula = rowkey + column family + column key + timestamp

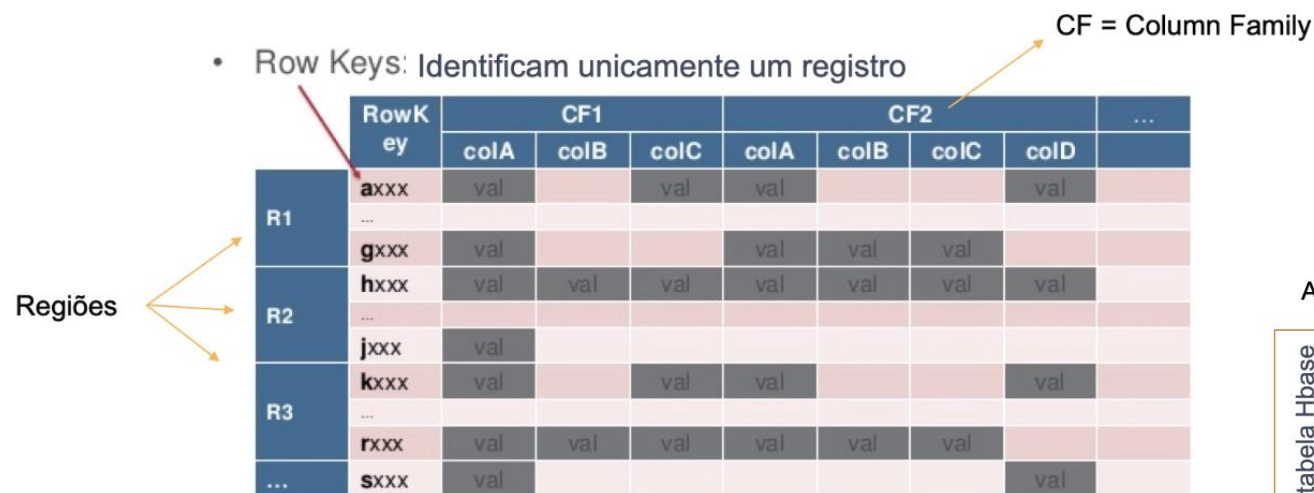
A proposta do Hbase é armazenar **PetaBytes** de dados!

Timestamp é uma sequência de caracteres que identifica quando um evento ocorre e normalmente com dados de hora e nível de fração de segundos.

FAMILIA DE COLUNAS

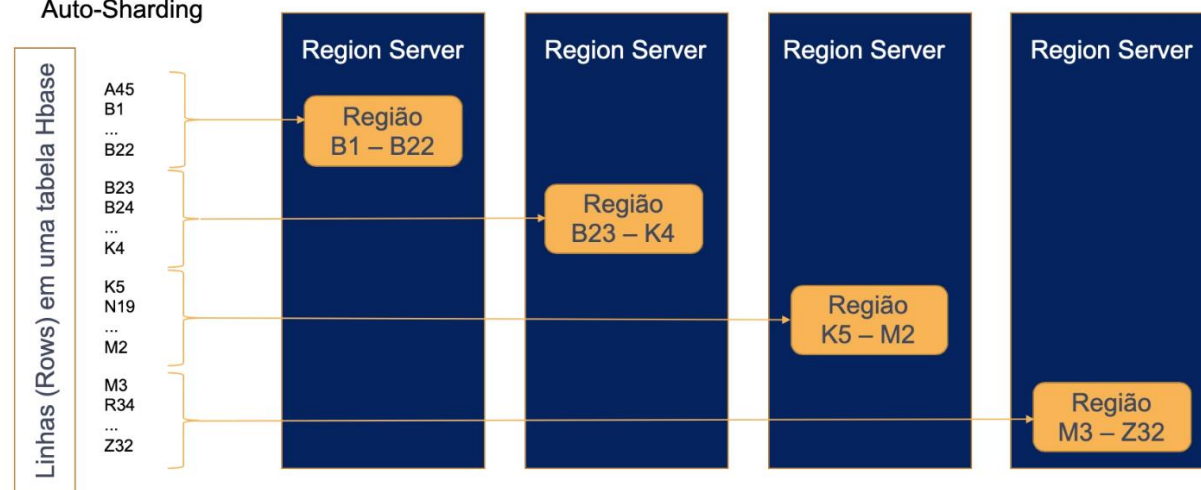


- Uma tabela Hbase é uma combinação de famílias de colunas e RowKeys.
- **Regiões** são partições da tabela que podem ser distribuídas através de um cluster de computadores.



Partições:

Auto-Sharding



ORIENTAÇÃO A COLUNA

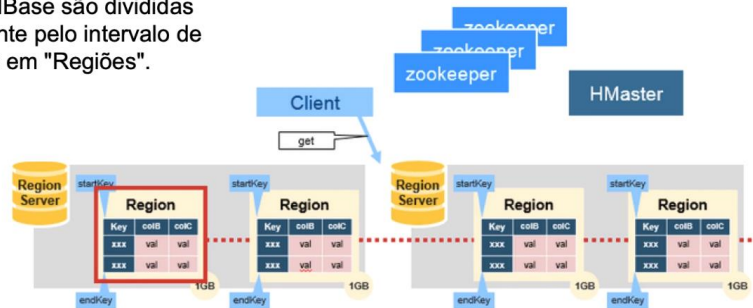


O HBase é orientado a coluna:

Rowkey	Clientes		Vendas	
ID Cliente	Nome Cliente	Cidade	ID Produto	Valor
1	Bob	SP	1002	500
2	Zico	RJ	1008	700

2 Famílias de colunas!

As tabelas HBase são divididas horizontalmente pelo intervalo de rowkeys em "Regiões".



Uma região contém todas as linhas da tabela entre a chave inicial e a chave final da região.

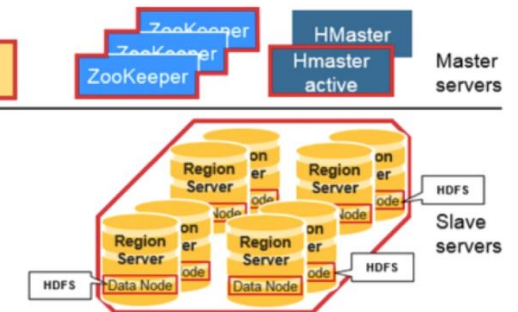
As regiões são atribuídas aos nós do cluster, chamados Region Servers. Um servidor de região pode atender cerca de 1.000 regiões.

Fisicamente, o HBase é composto por três tipos de servidores em uma arquitetura mestre/escravo. Servidores de Região servem os dados para leituras e gravações.

Ao acessar dados, os clientes se comunicam diretamente com os Region Servers. As operações de atribuição de região e DDL (criar e excluir tabelas) são tratadas pelo processo do HBase Master. O Zookeeper mantém o estado de cluster ativo.

O Hadoop DataNode armazena os dados que o Region Server está gerenciando. **Todos os dados do HBase são armazenados em arquivos HDFS.**

Os Region Servers são colocados com os DataNodes do HDFS, que permitem a localidade dos dados (colocando os dados perto de onde são necessários) para os dados servidos pelos Region Servers.

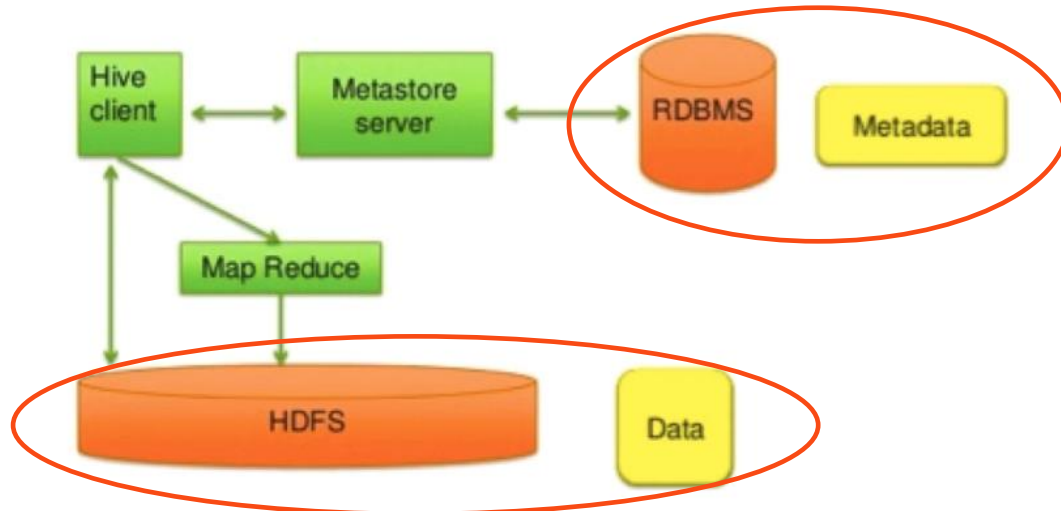


O NameNode mantém informações de metadados para todos os blocos de dados físicos que compõem os arquivos.

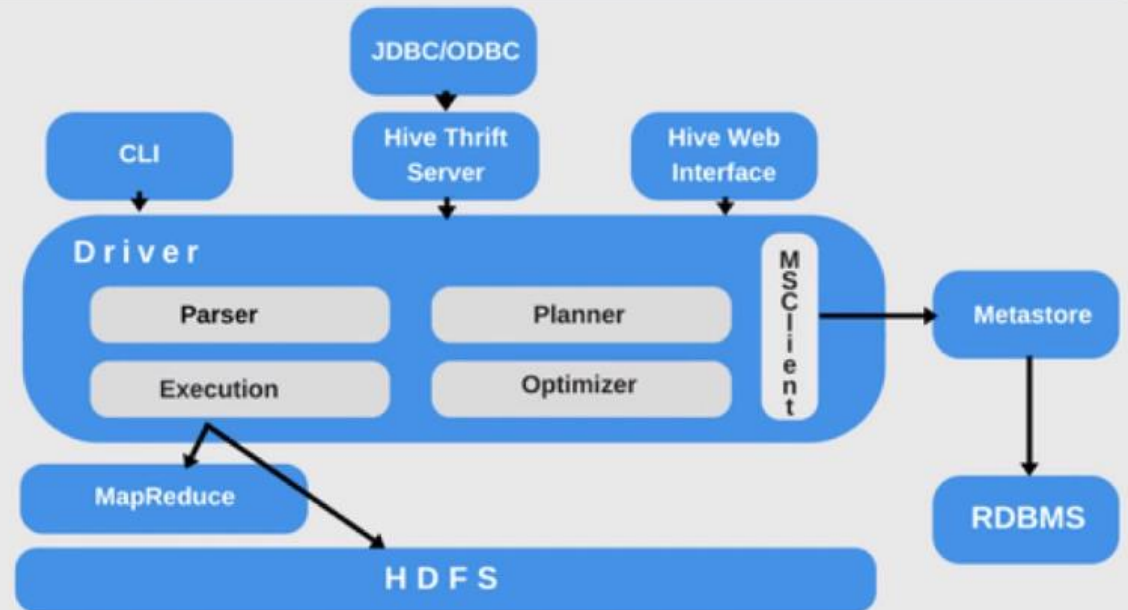
APACHE HIVE



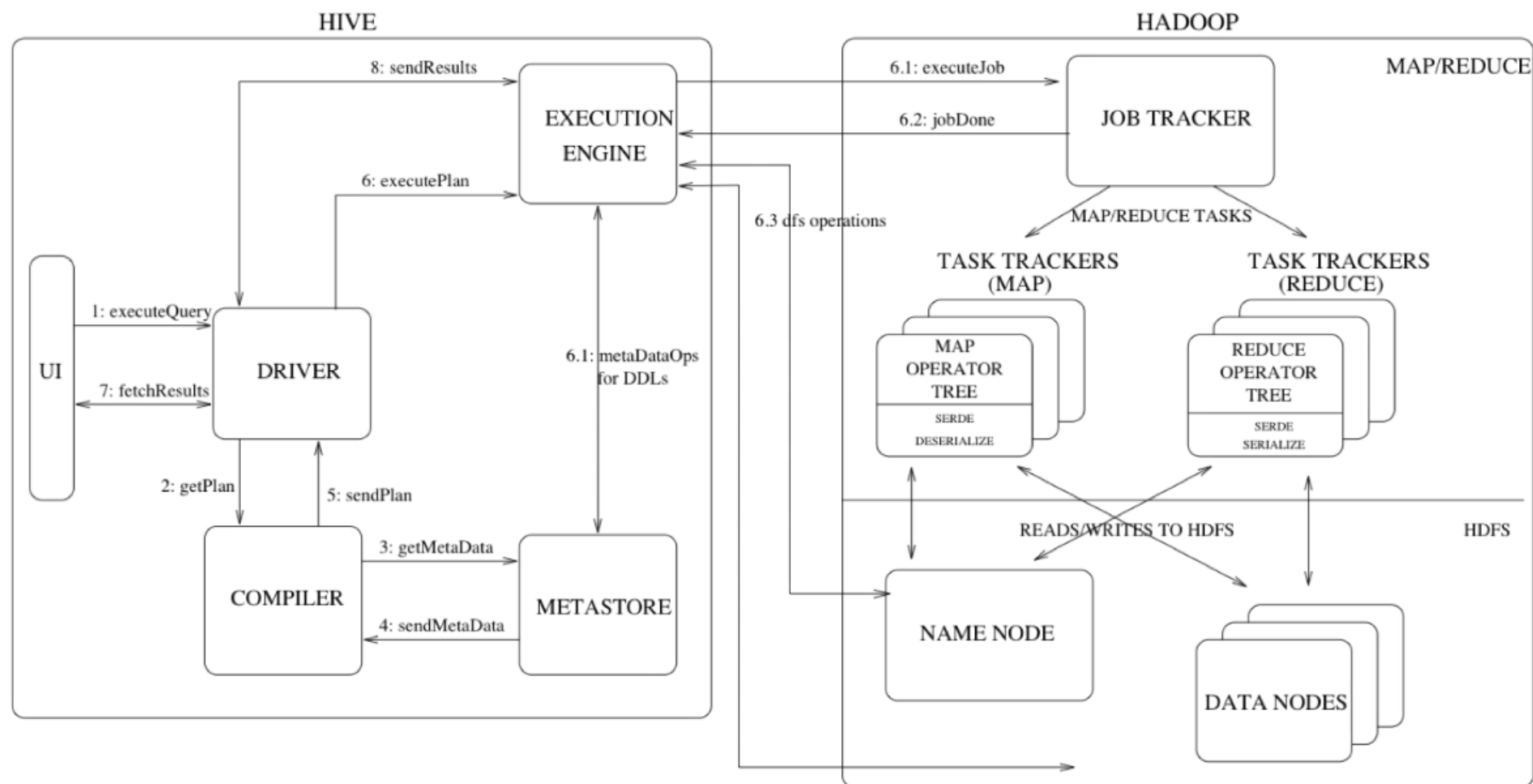
Banco de dados Relacional
(SQL on Hadoop - HQL)



Apache hive Architecture



APACHE HIVE



CONHECENDO O APACHE HIVE

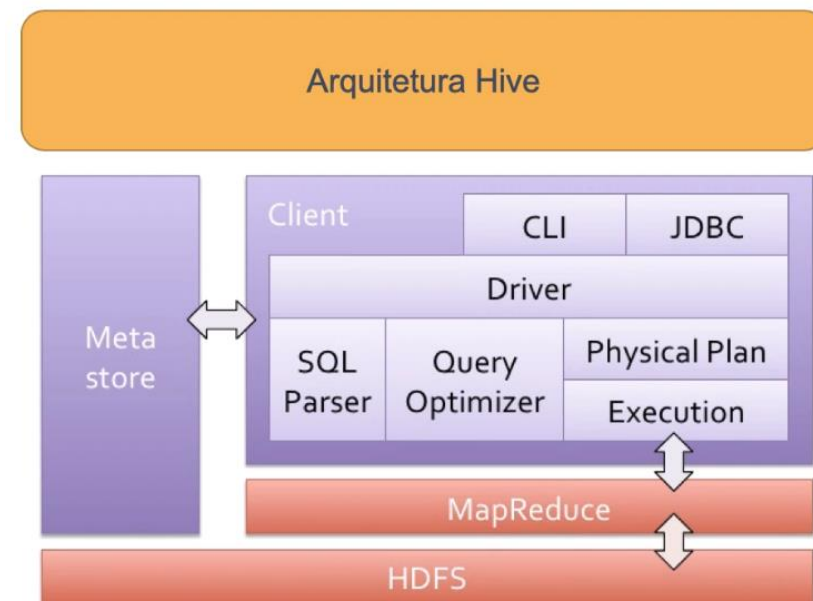
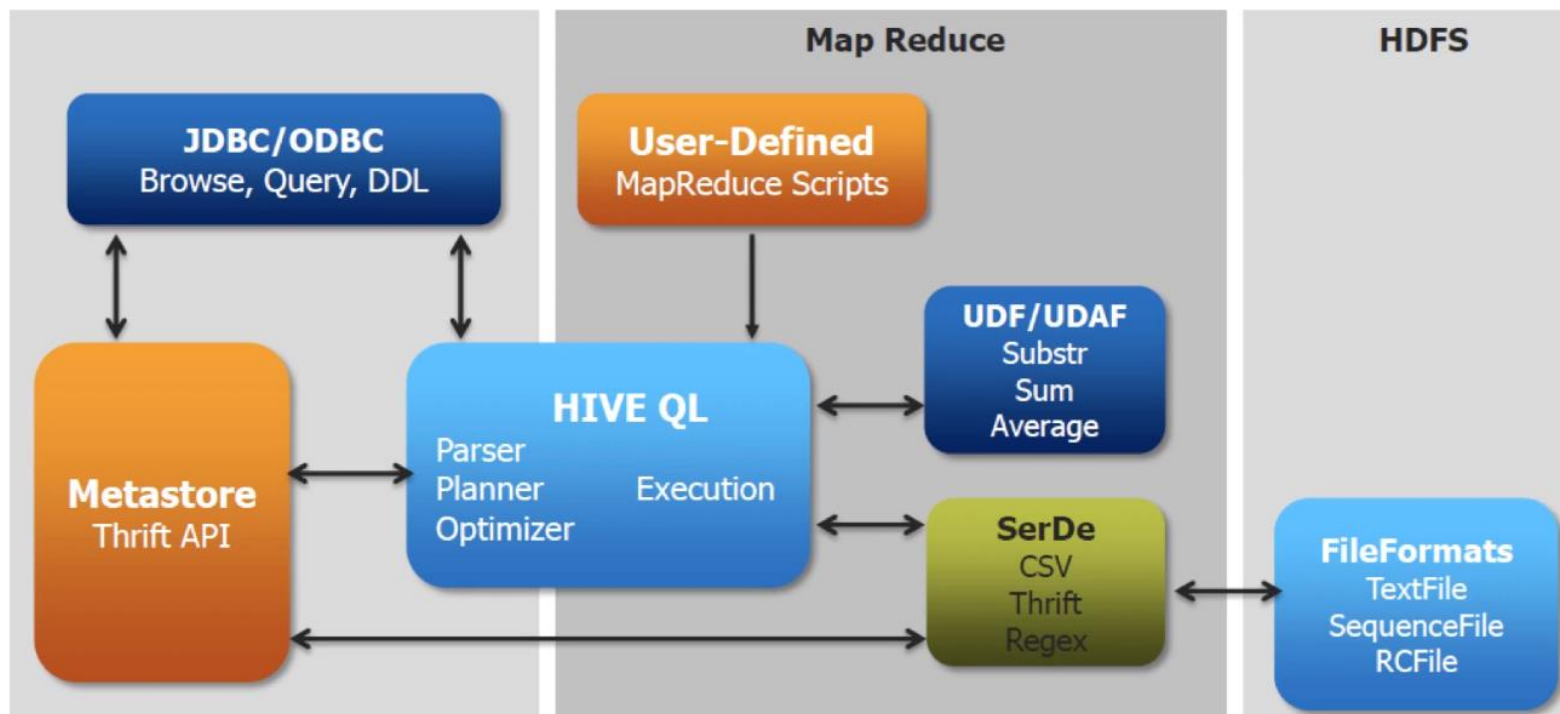


- O Apache Hive é uma ferramenta de infraestrutura de data warehousing para processar dados estruturados no Hadoop.
- Ele funciona acima do HDFS e permite a realização de consultas com uma linguagem similar ao SQL.
- Construído pelo Facebook em 2007.
- Baixo custo, escalabilidade, evita a dependência sobre custos de licença e manutenção anual.
- Operações de full Scan em grandes volumes de dados.
- O Hive utiliza o Hive Query Language
- SQL → Jobs MapReduce
- O Hive não é um banco de dados relacional para o dia a dia, não é para Online Transaction Processing (OLTP) (transações em tempo real), não é uma solução para consultas em tempo real e atualizações em nível de linha.

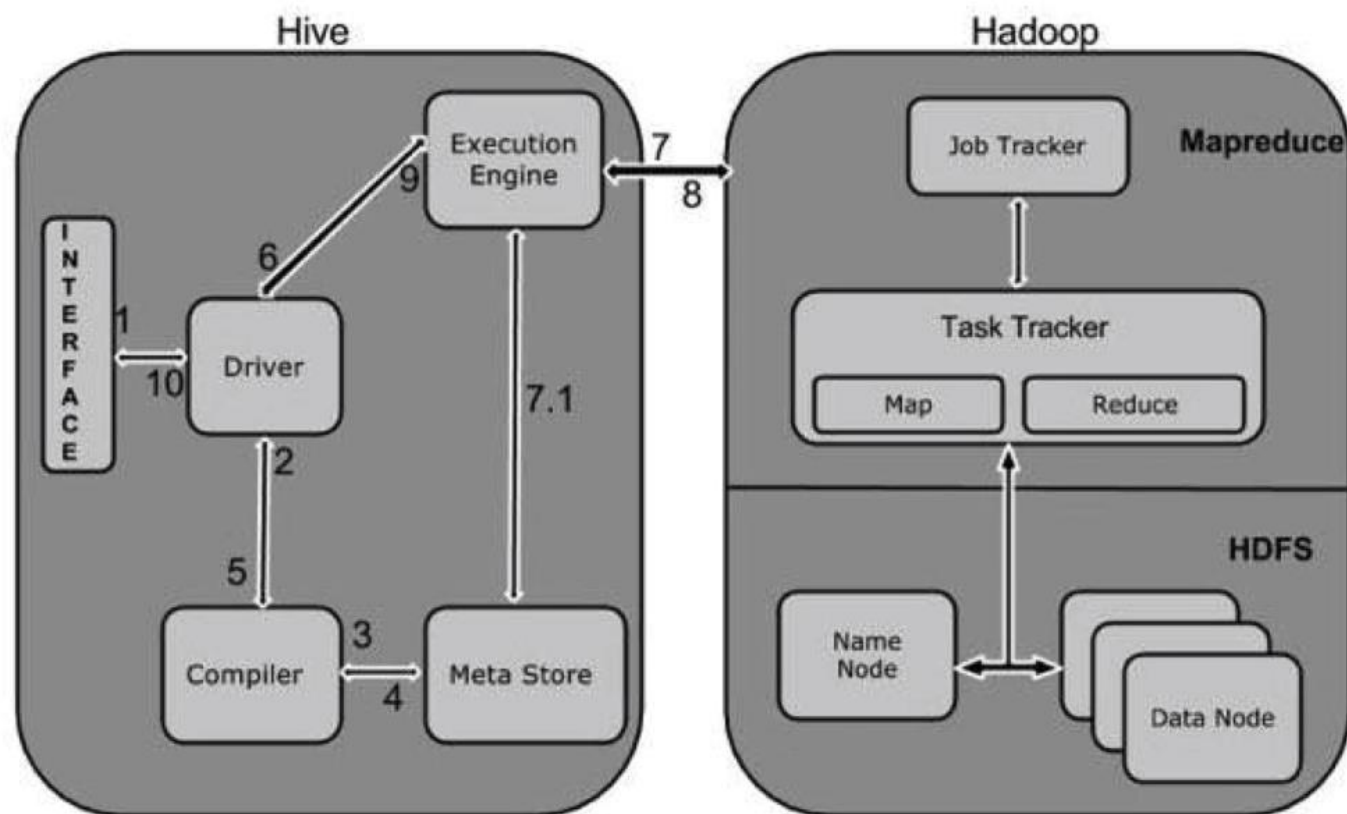
Usamos o Hive quando precisamos realizar consultas ou manipulações em grandes conjuntos de dados, tais como seleção de registros ou colunas, agregação, sumarização, contagem de elementos, filtros ou atualizações em massa.

Essas tarefas não precisam ser feitas em tempo real e o que queremos é obter insights a partir de grandes conjuntos de dados, Big Data.

ARQUITETURA DO APACHE HIVE



ARQUITETURA DO APACHE HIVE



Tipos de Dados

Tipos de Coluna
(Int, Smallint, Tinyint, Bigint)
(Char, Varchar)
Dates
Timestamp

Tipos Literais
(Ponto Flutuante e Decimal)

Tipo Nulo

Tipos Complexos
(Arrays, Maps, Structs)



THANKS