DIGITAL INNOVATION ONE

UNIVERSITY everis

Aceleração Global Dev #4 everis

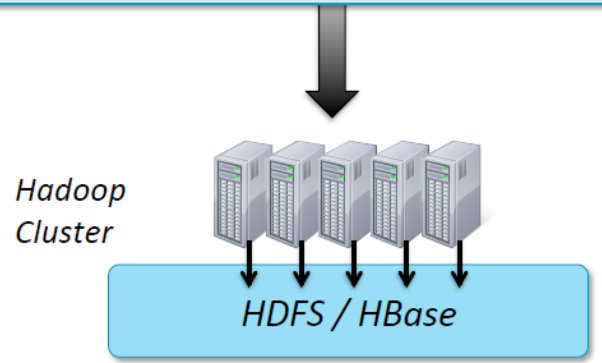
Apache Hive
Apache Impala

Vinícius Manini Bueno
Data Engineer

Introdução

Ambos, Impala e Hive, provêm SQL para consultas nos dados do HDFS e HBase

```
SELECT zipcode, SUM(cost) AS total
FROM customers
JOIN orders
ON (customers.cust_id = orders.cust_id)
WHERE zipcode LIKE '63%'
GROUP BY zipcode
ORDER BY total DESC;
```



Hive Definição

"Bringing this data closer to users is what inspired us to build Hive"

Ashish Thusoo

Software construído sobre o Apache Hadoop

Desenvolvido em 2007 pelo Facebook

A partir de 2008 Projeto Open Source

Fornece acesso semelhante ao SQL(HQL)

Hive Definição

HQL Hive Query Language

Hive é uma abstração em alto nível do MapReduce

- HQL : Linguagem de consulta do tipo sql;
- Tem semânticas e funções semelhantes às do SQL padrão no banco de dados relacional ;
- As consultas HQL são traduzidas implicitamente em um ou mais serviços de MapReduce, protegendo o usuário de programação muito avançada e consumo de tempo
- Gera jobs MapReduce ou Spark no cluster Hadoop;
- A linguagem de consulta do Hive pode ser executada em diferentes mecanismos de computação, como MapReduce, Tez e Spark;

Impala Definição

Impala é um motor SQL de alta performance

- Uma engine MPP (Massive Parallel Processing) open source para a execução de queries SQL em ambiente Hadoop;
- Inspirado no Projeto Google Dremel;
- Possui baixa latência (em milisegundos);
- Desenvolvido pela Cloudera em 2012 e agora é um top Project da Apache.

Impala é muito mais rápido!

- Oferece melhora de performance de 5x a 50x.
- Ideal para queries interativas e analise de dados.
- Muitas novidades são implementadas a todo o momento.

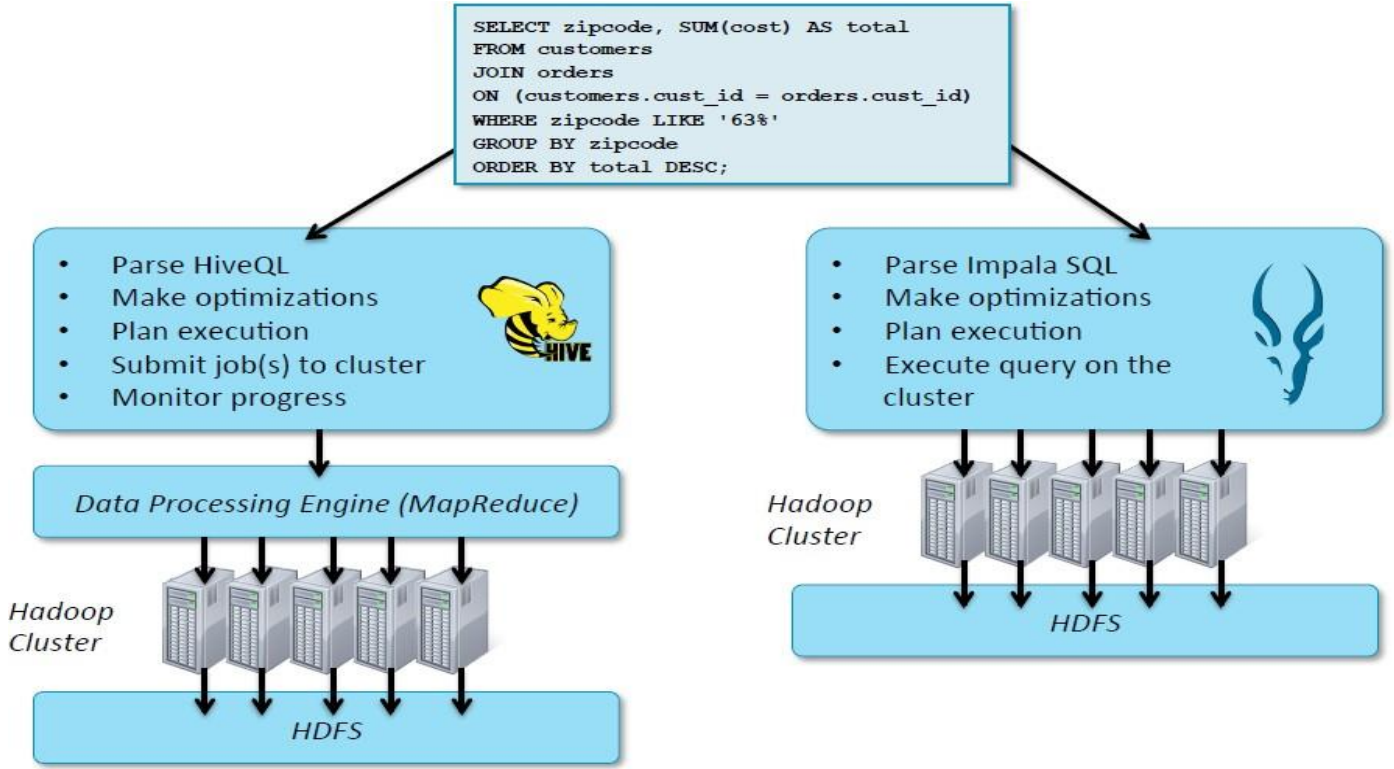
Hive x Impala

- O Impala não salva resultados intermediários no disco durante consultas demoradas. Dependendo da versão, o Impala cancela uma consulta em execução se algum host no qual essa consulta está executando falhar.

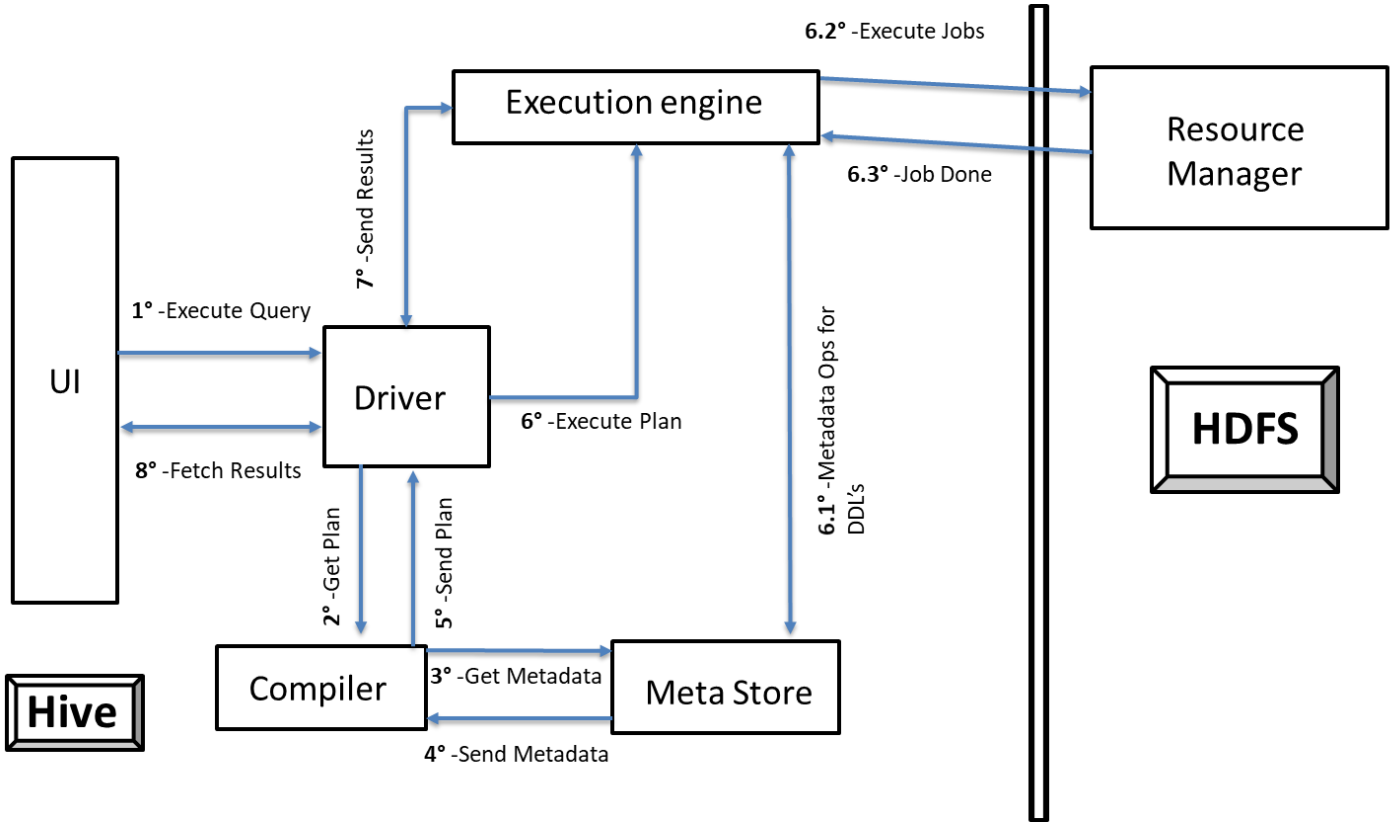
- Se uma aplicação tiver necessidades de processamento em lote de dados o hive pode ser a melhor opção.

- Se houver necessidade de processamento em tempo real de consultas ad-hoc em um subconjunto de dados, o Impala será a melhor opção.

Hive x Impala



Arquitetura Hive



DIGITAL INNOVATION ONE

Hive x Impala

UNIVERSITY everis

- O Impala é mais rápido que o Apache Hive, mas isso não significa que ele seja a única solução SQL para todos os problemas de big data.

- O Impala consome muita memória e não é executado de forma eficiente para operações de dados pesados, como joins, porque não é possível inserir tudo na memória

DIGITAL INNOVATION ONE

Comparando

UNIVERSITY everis

	Relational Database	Hive	Impala
Query language	SQL (full)	SQL (subset)	SQL (subset)
Update individual records	Yes	No	No
Delete individual records	Yes	No	No
Transactions	Yes	No	No
Index support	Extensive	Limited	No

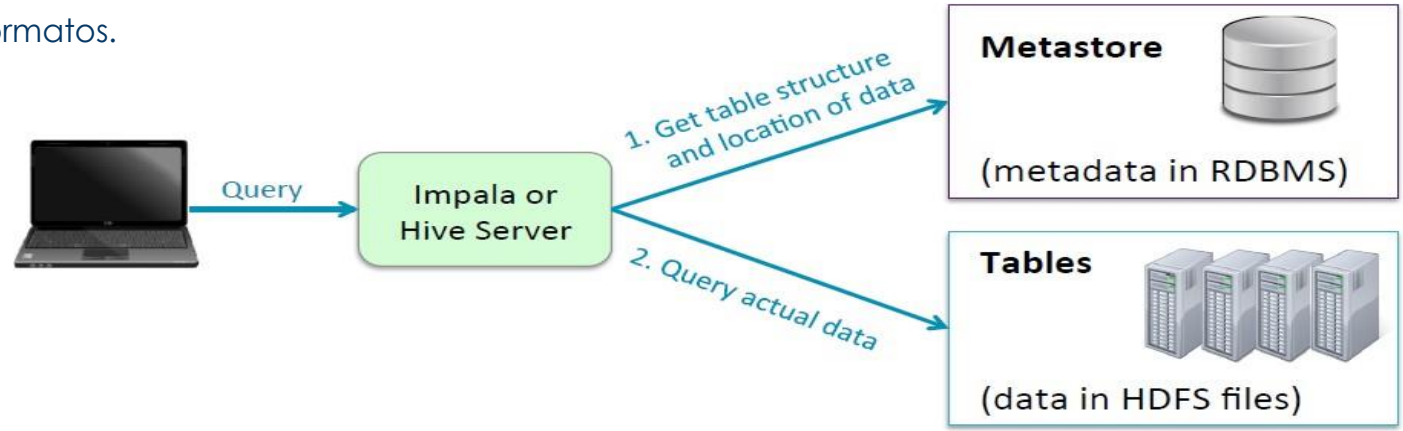
Metastore

Hive e Impala trabalham com O MESMO DADO

- Tabelas no HDFS, metadado e metastore

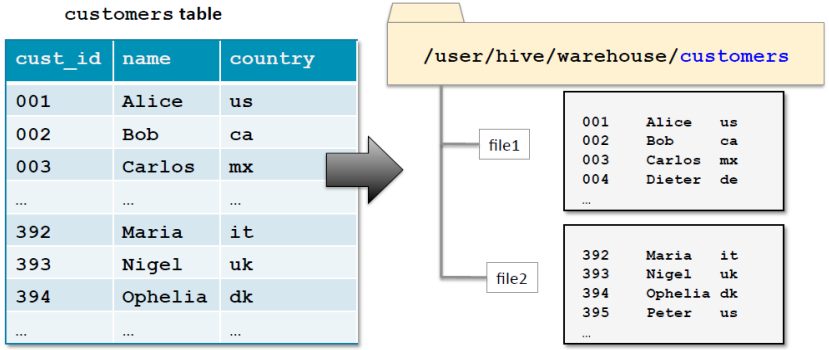
Queries são feitas em tabelas, assim como num banco de dados tradicional

- Uma tabela é simplesmente um diretório no HDFS contendo um ou vários arquivos;
- Caminho padrão: /user/hive/warehouse/<table_name>
- Suporte à diversos formatos.



Modelo de Dados

- Os dados são organizados em forma de tabelas e partições;
- Uma tabela é simplesmente um diretório no HDFS contendo um ou vários arquivos;




```
CREATE DATABASE IF NOT EXISTS treino;

DROP DATABASE IF EXISTS treino;

USE treino;
```

DIGITAL INNOVATION ONE

Tipos de Dados

UNIVERSITY everis

Numeric Types

- `TINYINT` (1-byte signed integer, from -128 to 127)
- `SMALLINT` (2-byte signed integer, from -32,768 to 32,767)
- `INT/INTEGER (4-byte signed integer, from -2,147,483,648 to 2,147,483,647)`
- `BIGINT` (8-byte signed integer, from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
- `FLOAT` (4-byte single precision floating point number)
- `DOUBLE` (8-byte double precision floating point number)
- `DOUBLE PRECISION` (alias for DOUBLE, only available starting with Hive 2.2.0)
- `DECIMAL`
  - Introduced in Hive 0.11.0 with a precision of 38 digits
  - Hive 0.13.0 introduced user-definable precision and scale
- `NUMERIC` (same as `DECIMAL`, starting with Hive 3.0.0)

DIGITAL INNOVATION ONE

Tipos de Dados

UNIVERSITY everis

Date/Time Types

- `TIMESTAMP` (Note: Only available starting with Hive 0.8.0)
- `DATE` (Note: Only available starting with Hive 0.12.0)
- `INTERVAL` (Note: Only available starting with Hive 1.2.0)

String Types

- `STRING`
- `VARCHAR` (Note: Only available starting with Hive 0.12.0)
- `CHAR` (Note: Only available starting with Hive 0.13.0)

Misc Types

- `BOOLEAN`
- `BINARY` (Note: Only available starting with Hive 0.8.0)

DIGITAL INNOVATION ONE

Database

UNIVERSITY everis

```
CREATE DATABASE loudacre;
```

```
CREATE DATABASE IF NOT EXISTS loudacre;
```

/user/hive/warehouse/loudacre.db

```
DROP DATABASE loudacre;
```

```
DROP DATABASE IF EXISTS loudacre;
```

DIGITAL INNOVATION ONE

Database

UNIVERSITY everis

```
CREATE TABLE tablename (colname DATATYPE, ...)
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY char
  STORED AS {TEXTFILE|SEQUENCEFILE|…}
```

- Default database:
  /user/hive/warehouse/*tablename*
- Named database:
  /user/hive/warehouse/*dbname*.db/*tablename*

DIGITAL INNOVATION ONE

Table

UNIVERSITY everis

```
CREATE TABLE jobs (
    id INT,
    title STRING,
    salary INT,
    posted TIMESTAMP
  )
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ',';
```

```
1,Data Analyst,100000,2013-06-21 15:52:03
```

DIGITAL INNOVATION ONE

Table

UNIVERSITY everis

```
CREATE TABLE jobs_archived LIKE jobs;
```

```
CREATE TABLE ny_customers AS
  SELECT cust_id, fname, lname
  FROM customers
  WHERE state = 'NY';
```

DIGITAL INNOVATION ONE

Table

UNIVERSITY everis

Criando uma tabela MANAGED

- Basicamente um arquivos criado pelo próprio no HDFS;
- Caso você drope a tabela, todo o dado posteriormente criado será eliminado no processo!

```
CREATE TABLE jobs (
    id INT,
    title STRING,
    salary INT,
    posted TIMESTAMP
  )
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  LOCATION '/loudacre/jobs';
```

Table

Criando uma tabela EXTERNAL

- Cria um metadado para acesso ao arquivo no HDFS;
- Nesse caso, se dropar a tabela, o dado permanecerá.

```
CREATE EXTERNAL TABLE adclicks
  ( campaign_id STRING,
    click_time TIMESTAMP,
    keyword STRING,
    site STRING,
    placement STRING,
    was_clicked BOOLEAN,
    cost SMALLINT)
  LOCATION '/loudacre/ad_data';
```

DIGITAL INNOVATION ONE

Table

UNIVERSITY everis

```
SHOW TABLES;
+---------------+
|   tab_name    |
+---------------+
| accounts      |
| employees     |
| job           |
| vendors       |
+---------------+
```

```
DESCRIBE jobs;
+--------+-----------+---------+
| name   | type      | comment |
+--------+-----------+---------+
| id     | int       |         |
| title  | string    |         |
| salary | int       |         |
| posted | timestamp |         |
+--------+-----------+---------+
```

DIGITAL INNOVATION ONE

Table

UNIVERSITY everis

```
SHOW CREATE TABLE jobs;
+----------------------------------------------------+
| CREATE TABLE default.jobs                          |
|   id INT,                                          |
|   title STRING,                                    |
|   salary INT,                                      |
|   posted TIMESTAMP                                 |
| )                                                  |
| ROW FORMAT DELIMITED FIELDS TERMINATED BY ','      |
...
```

Formatos de arquivos

Ao se criar uma tabela, é possível que se determine o formato que ela terá

```
CREATE TABLE tablename (colname DATATYPE, ...)
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY char
   STORED AS format
```

Formatos de arquivos

```
CREATE TABLE order_details_parquet (
    order_id INT,
    prod_id INT)
  STORED AS PARQUET;
```

- É um formato colunar desenvolvido pela Cloudera e Twitter;
- Reduz o espaço de armazenamento;
- Aumenta a performance;
- Mais eficiente ao adicionar muitos registros de uma vez;
- A melhor escolha para acesso a dados colunares.

DIGITAL INNOVATION ONE

Formatos de arquivos

UNIVERSITY everis

DIGITAL INNOVATION ONE

Comparação

UNIVERSITY everis

BIG DATA FORMATS COMPARISON

	Avro	Parquet	ORC
Schema Evolution Support			
Compression			
Splitability			
Most Compatible Platforms	Kafka, Druid	Impala, Arrow Drill, Spark	Hive, Presto
Row or Column	Row	Column	Column
Read or Write	Write	Read	Read

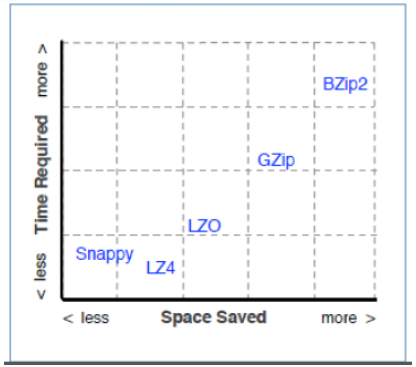
Apache orc

AVRO

Parquet

Formatos de arquivos

- Pode melhorar a performance significativamente;
- Melhora o trafego de dados na rede;
- Algoritmos mais agressivos ocupam menos espaço, mas pode ser lentos para leitura;
- Algoritmos menos agressivos, ocupam menos espaços, mas geralmente são mais rápidos;
- A implementação do algoritmo de compressão chama-se codec;
- Muitos codecs são utilizados com Hadoop;
- Cada codec tem características de performance diferentes;
- LZ4 e Snappy são os mais rápidos;
- Impala suporta Snappy, mas não suporta LZ4.



Tipos de tabelas

EXTERNAL TABLE x MANAGED TABLE

EXTERNAL TABLE

- Hive assume que não gerencia os dados;
- DROP apaga somente os metadados ;
- Use tabelas externas quando os arquivos já estiverem presentes ou em locais remotos;
- Use tabelas externas quando os arquivos devem permanecer mesmo se a tabela for descartada;

MANAGED TABLE

- Os dados são armazenados no diretório warehouse do Hive;
- Localizado em /hive/warehouse/
- Sempre que uma tabela for alterada, os dados também serão alterados;
- Use tabelas gerenciadas quando o Hive deve gerenciar o ciclo de vida da tabela ou ao gerar tabelas temporárias.

Criando tabelas

External Table

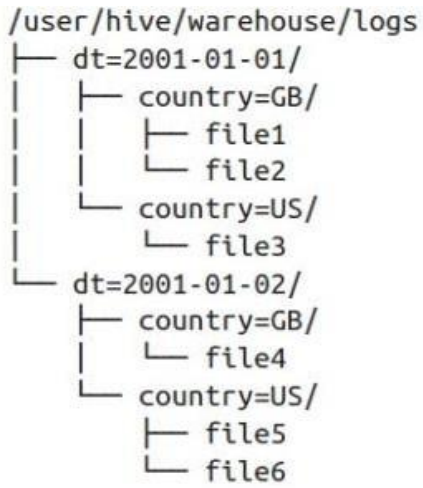
```
CREATE EXTERNAL TABLE if not exists  beca.jobs
(
    id int COMMENT 'codigo de identificacao',
    descricao int COMMENT 'nome do job',
    salario float COMMENT 'salario pago pelo  job'
) COMMENT 'tabela de descricao de jobs'
ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\;'
STORED AS TEXTFILE
LOCATION '/beca/jobs';
```

Managed Table

```
CREATE TABLE if not exists beca.jobs (
    id int COMMENT 'codigo de identificacao',
    descricao int COMMENT 'nome do job',  salario
    float COMMENT 'salario pago pelo
job'
) COMMENT 'tabela de descricao de jobs'
ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\;'
STORED AS TEXTFILE;
```

Particionamento

A partição determina como os dados são armazenados



Particionamento

- Não particione pouco demais: o particionamento por gênero por exemplo só cria duas partições ("masculino" e "feminino"), reduzindo a latência no máximo pela metade
- Não particione demais: Partição demais causa muita carga sobre namenode do cluster porque ele precisa manipular o grande número de diretórios.

```
CREATE TABLE tab_part

(viewTime INT, userid BIGINT,page_url STRING, referrer_url STRING,ip STRING)

COMMENT 'Essa tabela é particionada'

PARTITIONED BY(dt STRING, country STRING)

STORED AS TEXTFILE;
```

Referências úteis

- ✓ http://hive.apache.org/
- ✓ https://cwiki.apache.org/confluence/display/Hive/Home
- ✓ https://cwiki.apache.org/confluence/display/Hive
- ✓ https://cwiki.apache.org/confluence/display/Hive/GettingStarted
- ✓ https://impala.apache.org/

DIGITAL
INNOVATION
ONE

Dúvidas?

Apache Hive
Apache Impala