

Aula 3

Big Data

Prof. Luis Henrique Alves Lourenço

1

Conversa Inicial

2

Integração com bancos de dados

- Hive
- Sqoop
- HBase
- Cassandra e MongoDB
- Drill, Phoenix e Presto

3

Hive

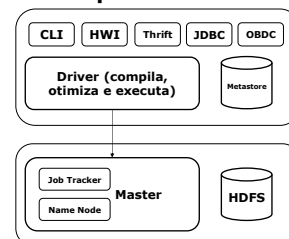
4

Hive

- *Data warehouse* projetado para o Hadoop
- HiveQL
- OnLine Analytical Processing (OLAP)
- Opera sobre MapReduce ou Tez
- Integra extensões customizadas e programas externos

5

Arquitetura Hive



6

HiveQL

- **Dialeto SQL**
- **Database: catálogo de tabelas**
- **Databases e tabelas são armazenados em uma árvore de diretórios**
- **Otimiza consultas por meio do MapReduce**
- ***Schema on read***

7

Integração com bancos de dados relacionais

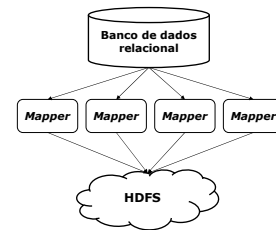
8

Sqoop

- **Transferência de dados entre Hadoop e aplicações externas**
- **Utiliza MapReduce ou Hive**
- **Grande flexibilidade**

9

Fluxo de importação de dados do Sqoop



10

Exemplo de importação

- `sqoop import --connect jdbc:mysql://localhost/foobar --driver com.mysql.jdbc.Driver --table foo`

11

Exemplo de importação para o Hive

- `sqoop import --connect jdbc:mysql://localhost/foobar --driver com.mysql.jdbc.Driver --table foo --hive-import`

12

Importação incremental

- Mantêm os dados no Hadoop sincronizados com o banco de dados relacional

13

Exemplo de exportação a partir do Hive

```
sqoop export --connect  
jdbc:mysql://localhost/foobar --driver  
com.mysql.jdbc.Driver --table exported_foo  
--export-dir /app/hive/warehouse/foo  
--input-fields-terminated-by '\0001'
```

14

Bancos de dados não relacionais

15

Teorema CAP

- Consistência
- Disponibilidade
- Tolerância a falhas (partição)

16

NoSQL

- Escalabilidade horizontal
- Dados estruturados, semiestruturados, não estruturados
- Complexidade de bancos de dados tradicionais

17

Bancos de dados de esquema chave-valor

- Modelo mais simples
- Dados não estruturados
- Os registros são armazenados como *arrays* associativos
- Flexibilidade de tipos
- Discretamente ordenado

18

Banco de dados baseado em documentos

- Extensão do modelo de esquema chave-valor
- Dados semiestruturados
- XML, YAML, JSON, BSON
- Esquemas altamente flexíveis
- Capaz de consultar campos dos documentos

19

Banco de dados baseado em colunas

- Famílias de colunas
- Dados semiestruturados
- Dados esparsos
- Versionamento de registros

20

Banco de dados baseado em grafos

- Teoria dos grafos: nós e relações
- Dados semiestruturados
- As relações são mais importantes que os dados
- Pode utilizar outro tipo de armazenamento nos nós

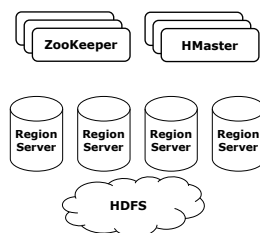
21

HBase

- Não implementa SQL
- API Java baseado em CRUD
- Baseado no MapReduce

22

Arquitetura HBase



23

Bancos NoSQL externos

24

Cassandra

- Banco de dados NoSQL distribuído
- Modelo de armazenamento chave-valor
- Linguagem CQL semelhante à do SQL
- Disponibilidade e tolerância a falhas
- Consistência eventual
- Integração nativa com o Hadoop

25

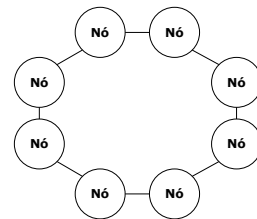
- Diferenças com bancos relacionais
 - Suporta transações leves que suportam propriedades ACID parcialmente
 - Não efetua JOIN
 - Não possui chaves estrangeiras
 - Não possui integridade referencial

26

- Keyspace
 - Tabelas
 - *Materialized view*
 - *Agregate*
 - Tipos e funções customizados

27

Arquitetura Cassandra



28

MongoDB

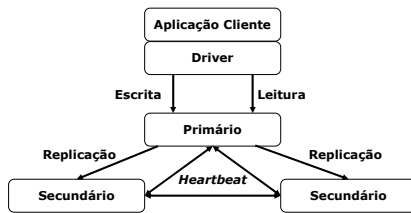
- Banco de dados NoSQL distribuído
- Modelo de dados baseado em documentos
- JSON
- Consistência e tolerância a falhas

29

- Coleções
 - Documentos
- Esquemas e modelagem dos dados
 - Estratégias de modelagem
 - ✓ Modelos de dados embutidos
 - ✓ Modelos de dados normalizados

30

Conjuntos de replicação



Particionamento

- Nós secundários não aumentam a escalabilidade do *cluster*, mas a disponibilidade
- Múltiplos conjuntos de replicação
- *Cluster*
- Partição (*shard*)
- Servidores de configuração

Motores de consulta SQL

Drill

- Modelo de dados em formato JSON
- Suporte à sintaxe do padrão SQL:2003
- Capaz de operar com ferramentas de BI
- Extensões ao SQL
- Múltiplas fontes de dados (NoSQL e sistemas de arquivos)

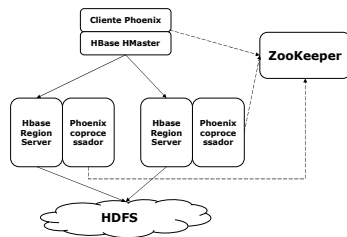
Drillbit

- Ambiente de execução distribuída
- Responsável por aceitar e processar as consultas e retornar resultados
- Pode ser instalado nos nós de diversos ambientes distribuídos, como o Hadoop
- Utiliza o ZooKeeper para coordenar o conjunto de serviços

Phoenix

- Consultas SQL para o HBase
- Pode ser utilizado para OLTP
- Oferece transações com propriedades ACID
- Possui integração com MapReduce, Spark, Hive e Flume

Arquitetura Phoenix



37

Presto

- Consultas SQL para OLAP
- Compatível com ANSI SQL
- Projetado pelo Facebook
- Múltiplas fontes de dados (NoSQL, bancos relacionais, sistemas de arquivos)

38

Arquitetura Presto

- Nó coordenador
- Nós trabalhadores
- Esquemas e referências de dados são armazenados em catálogos

39