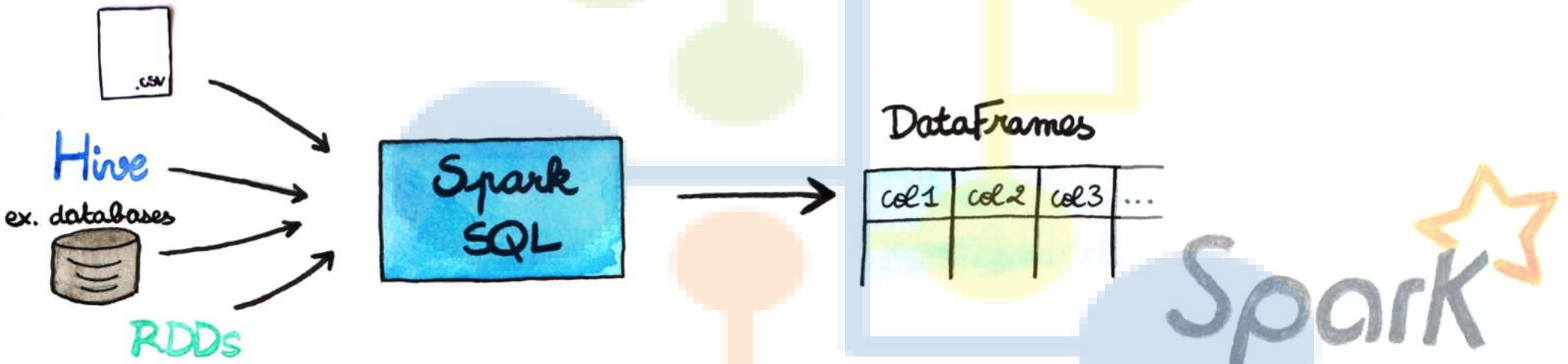
Big Data Real-Time Analytics com Python e Spark

Seja muito bem-vindo(a)!

Apache Spark SQL

Apache Spark SQL



Apache Spark SQL

O que vamos estudar neste capítulo?

- Computação em Nuvem e em Cluster
- Linguagem SQL
- Spark SQL com RDD's
- Spark SQL com Dataframes
- Spark SQL com Arquivos CSV
- Spark SQL com Arquivos JSON
- Spark SQL com Bancos de Dados Relacionais
- Spark SQL com Bancos de Dados Não-Relacionais
- Tabelas Temporárias com Spark SQL
- Como Construir um Cluster Spark em Nuvem

Computação em Nuvem e em Cluster

Computação em Nuvem e em Cluster

Computação em Nuvem e em Cluster

Não temos como processar e armazenar Big Data em apenas uma máquina.

Computação em Nuvem e em Cluster

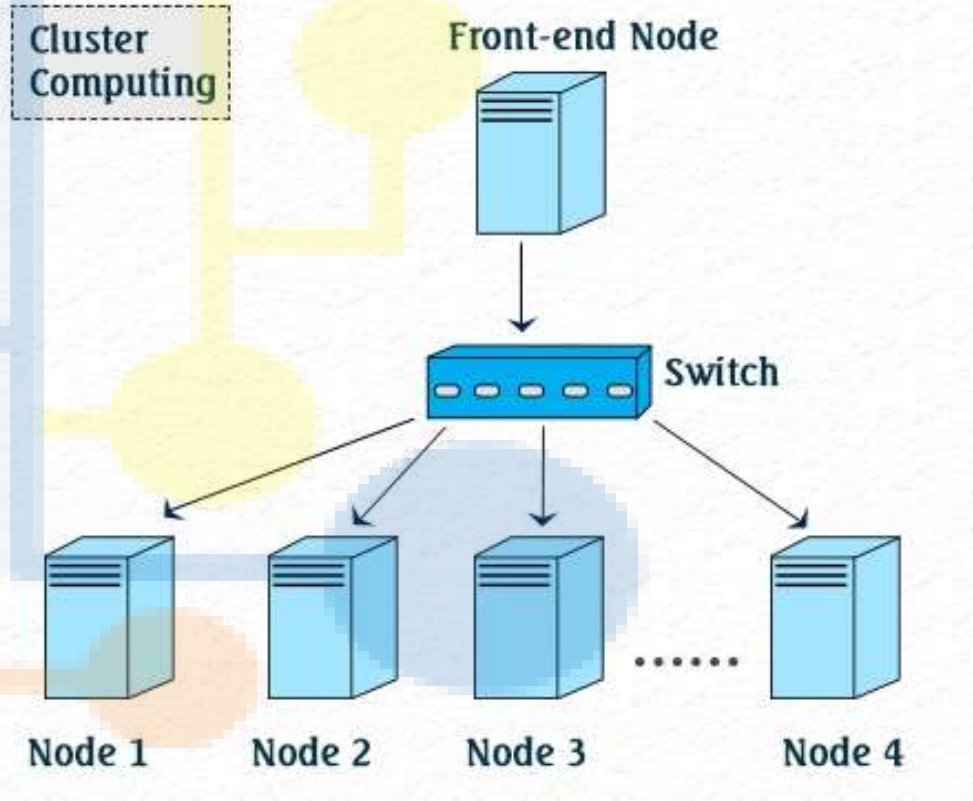
E a solução criada para poder armazenar e processar todo esse volume de dados, é a Cloud Computing. O conceito de Cloud Computing ou Computação em Nuvem é relativamente simples.

Computação em Nuvem e em Cluster

Computação em Nuvem e em Cluster

Um cluster é um sistema que compreende 2 ou mais computadores (chamados nodes) que trabalham em conjunto para executar aplicações ou realizar outras tarefas.

Um cluster oferece maior confiabilidade, distribuição de carga e confiabilidade



Computação em Nuvem e em Cluster

Computação em Nuvem e em Cluster

Computação em Nuvem e em Cluster

Por que não usar máquinas de baixo custo, ao invés de super computadores, e depois unir essas máquinas de baixo custo em clusters, de modo que pudéssemos criar um único sistema com milhares de máquinas, milhares de CPU's e muita, muita memória RAM?

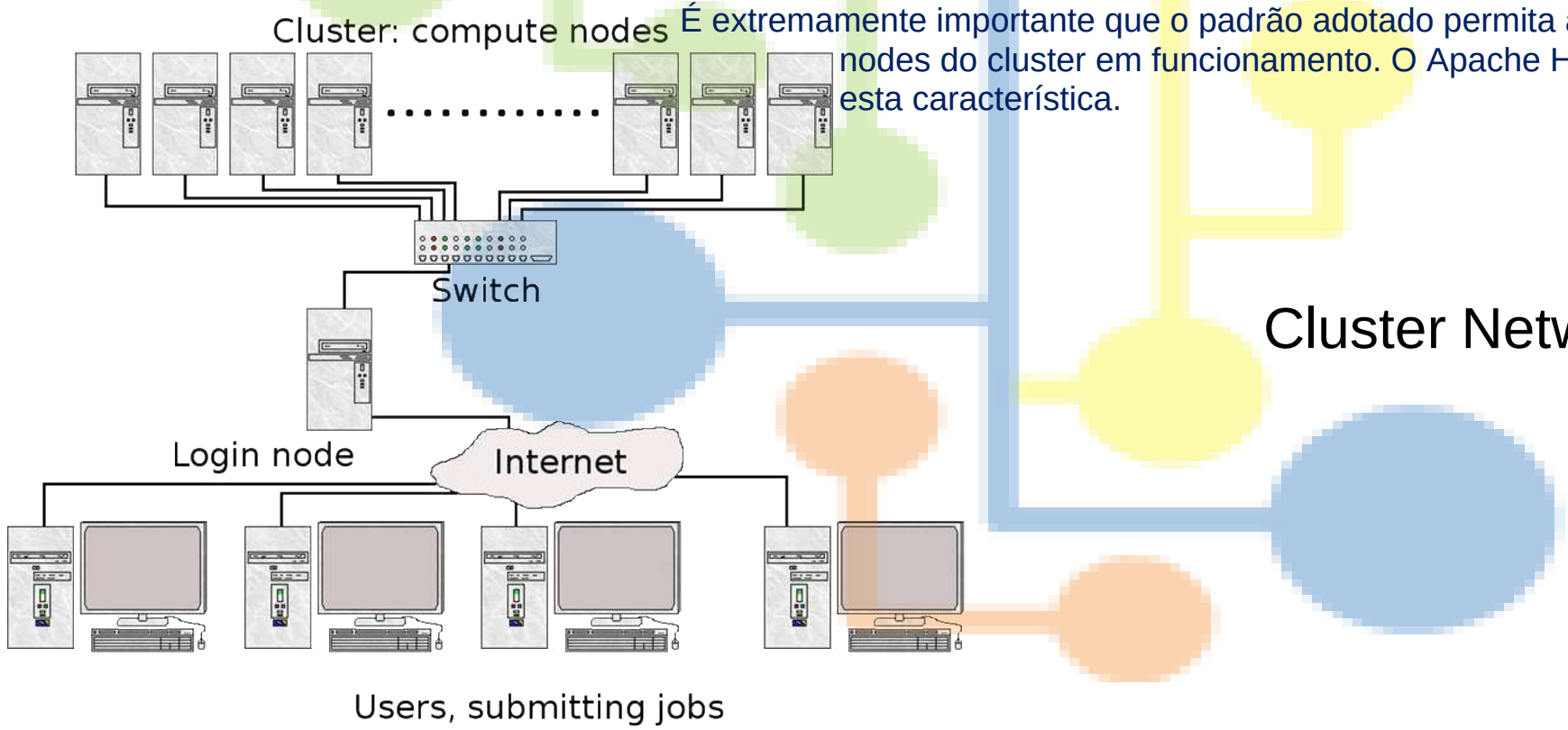
Computação em Nuvem e em Cluster

Google:

1 a 5% dos HD's vão falhar por ano e cerca de 0,2% de falhas com memória.

Computação em Nuvem e em Cluster

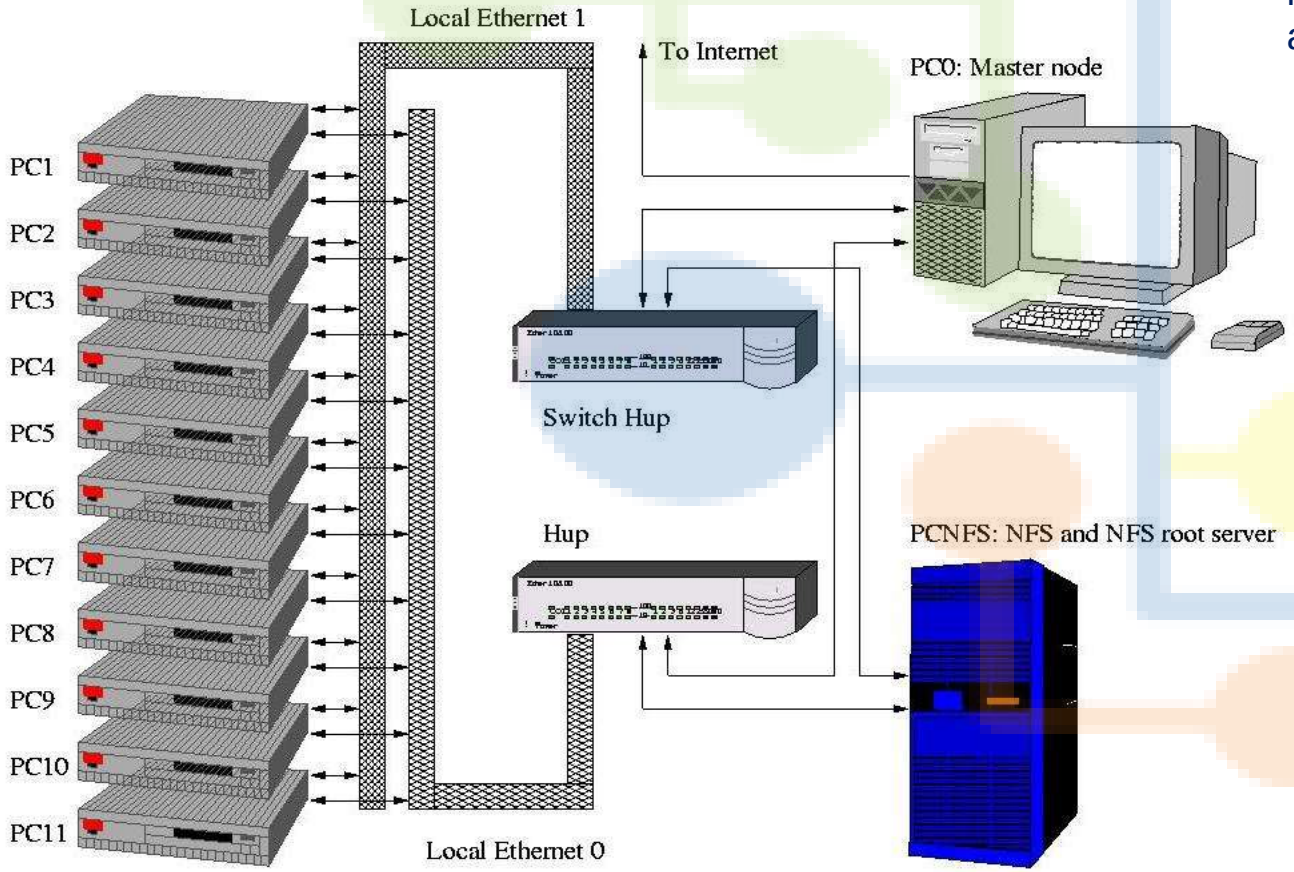
A comunicação em rede pelos computadores é outro problema do cluster. Os nodes do cluster devem ser interconectados, preferencialmente por uma tecnologia de rede conhecida, para fis de manutenção e controle de custo. O melhor exemplo é a rede ethernet (LAN), que pode sr utilizada para interconectar as máquinas em um cluster. É extremamente importante que o padrão adotado permita a inclusão ou remoção de nodes do cluster em funcionamento. O Apache Hadoop por exemplo possui esta característica.



Cluster Network

Computação em Nuvem e em Cluster

Os nodes podem ser compostos por computadores de configuração simples, que, juntos configuram um sistema robusto o suficiente para dar conta de determinadas aplicações. Tal fato propicia a diminuição de seu custo.



Cluster Network

Computação em Nuvem e em Cluster

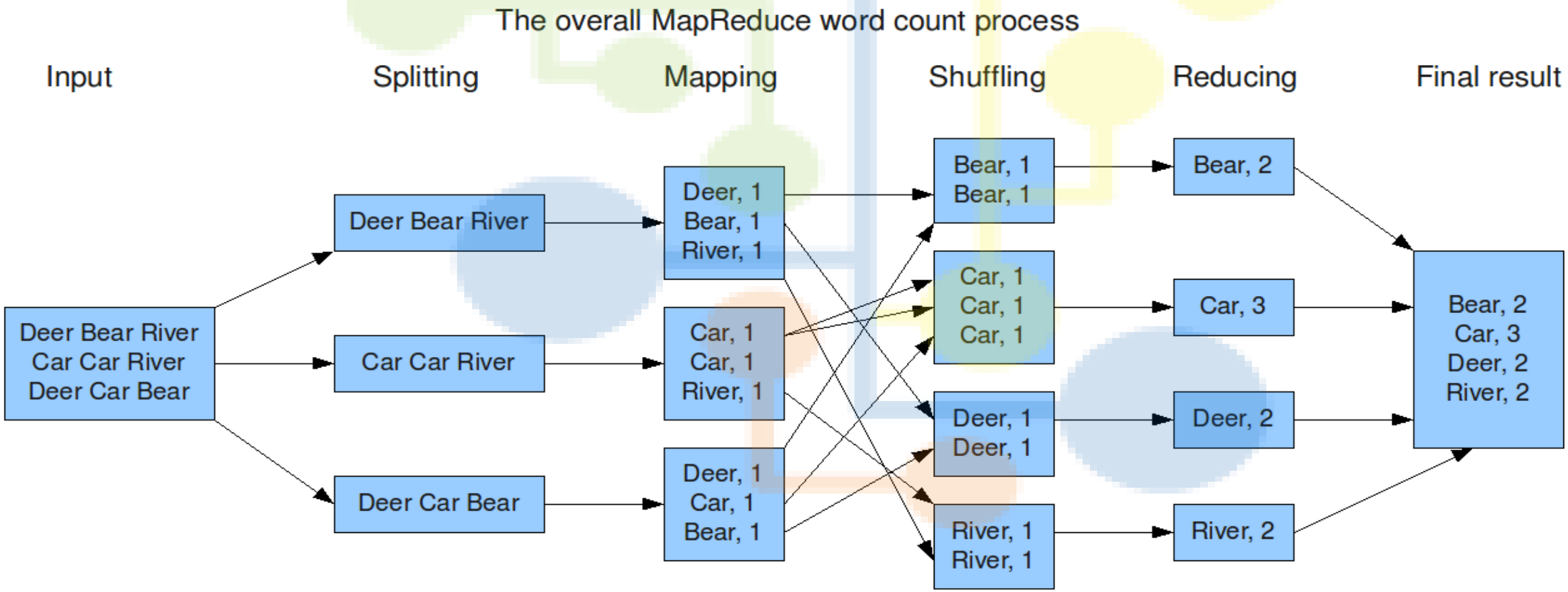
Ok, resolvemos o problema de armazenamento e processamento, distribuindo os dados por clusters (conjuntos de computadores) espalhados por datacenters em todo mundo e oferecendo este serviço pela internet!

Computação em Nuvem e em Cluster

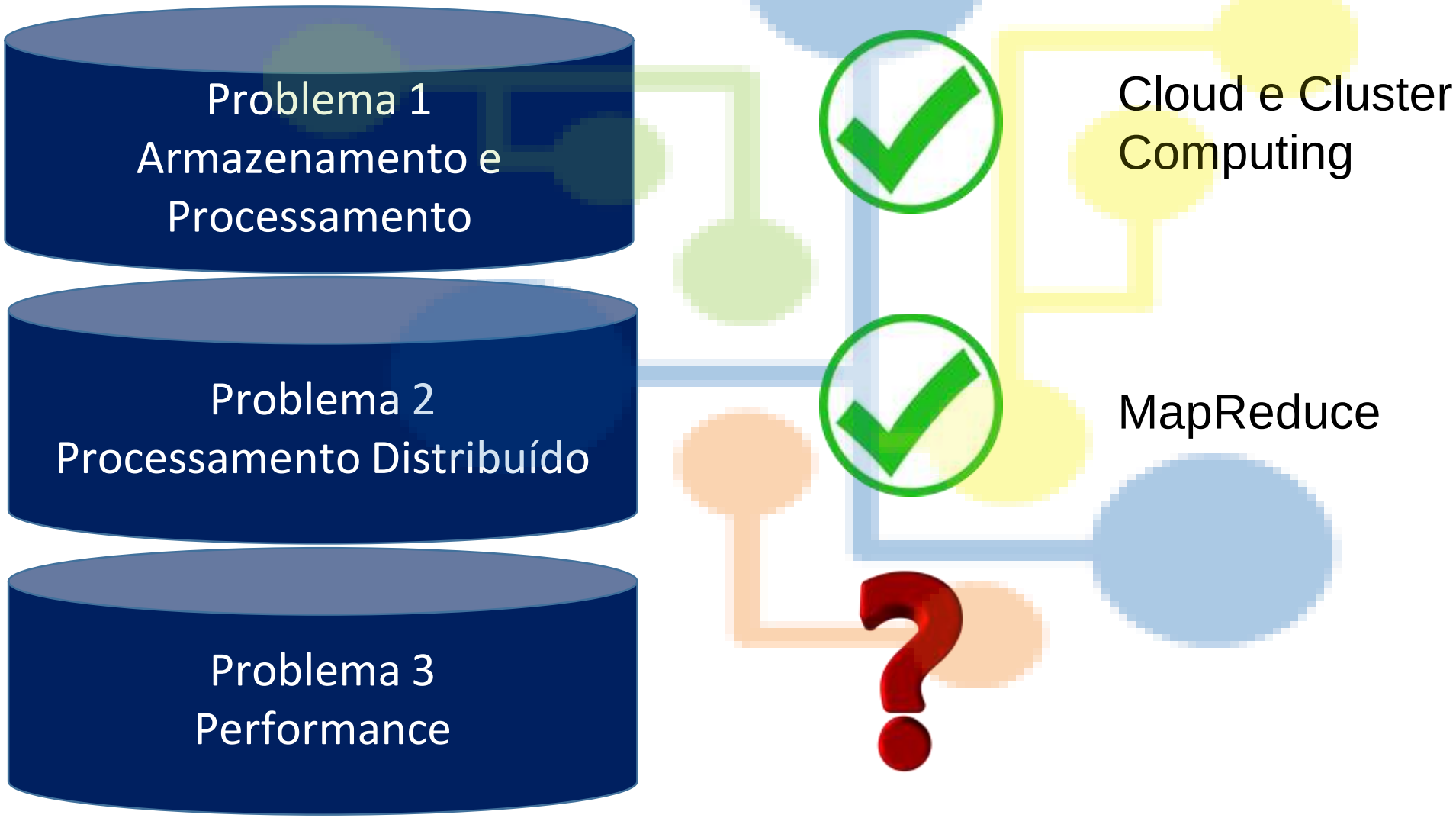
Mas como vamos processar esses dados distribuídos?

Computação em Nuvem e em Cluster

MapReduce é o paradigma de programação criado pela Apache Foundation para resolver o problema de processamento em cluster. O conceito por trás do MapReduce é o de dividir para conquistar.



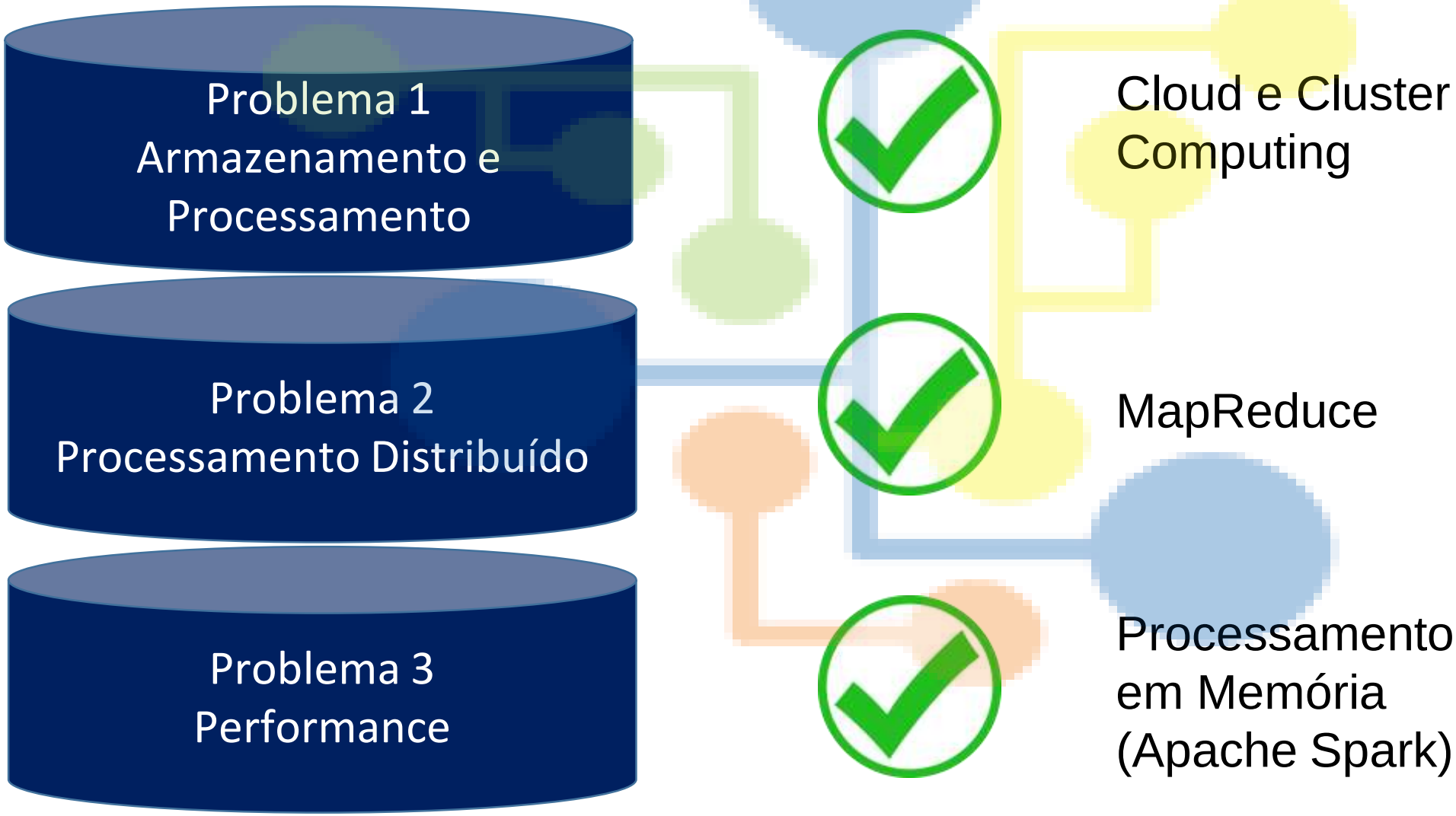
Computação em Nuvem e em Cluster



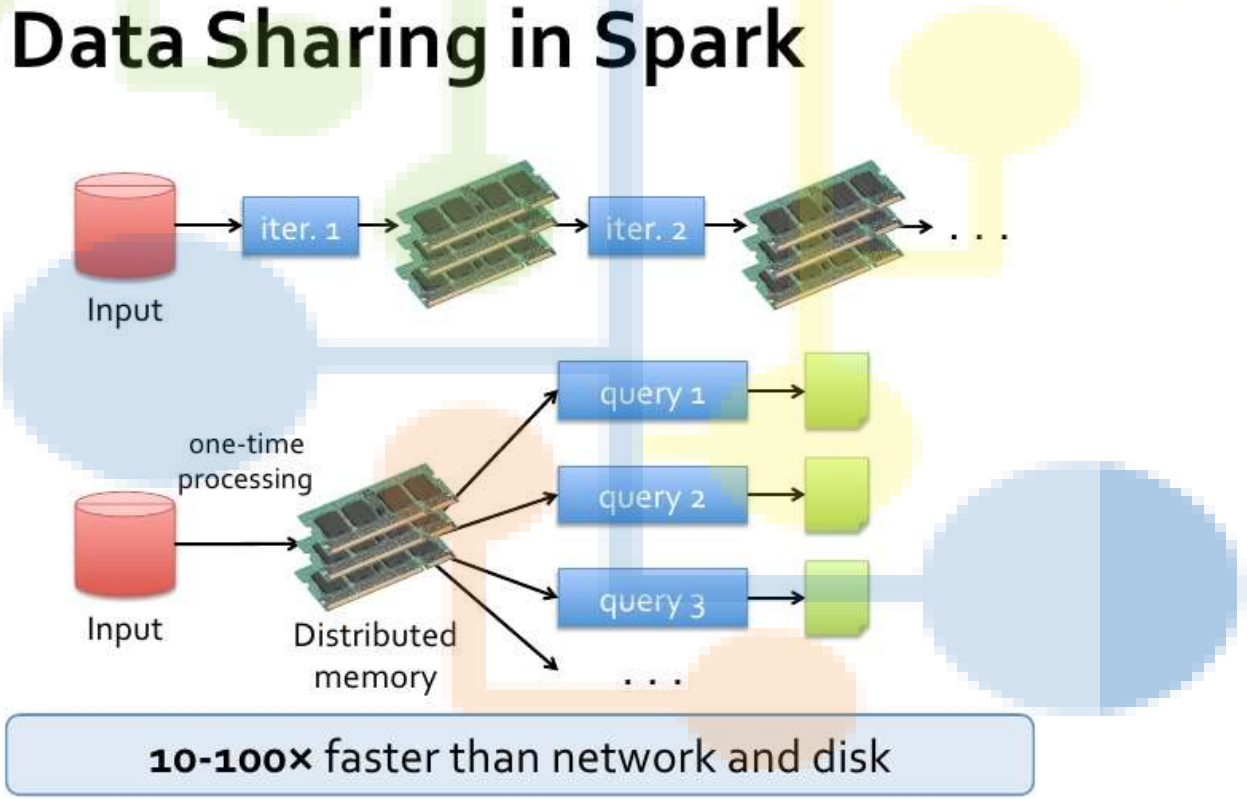
Computação em Nuvem e em Cluster

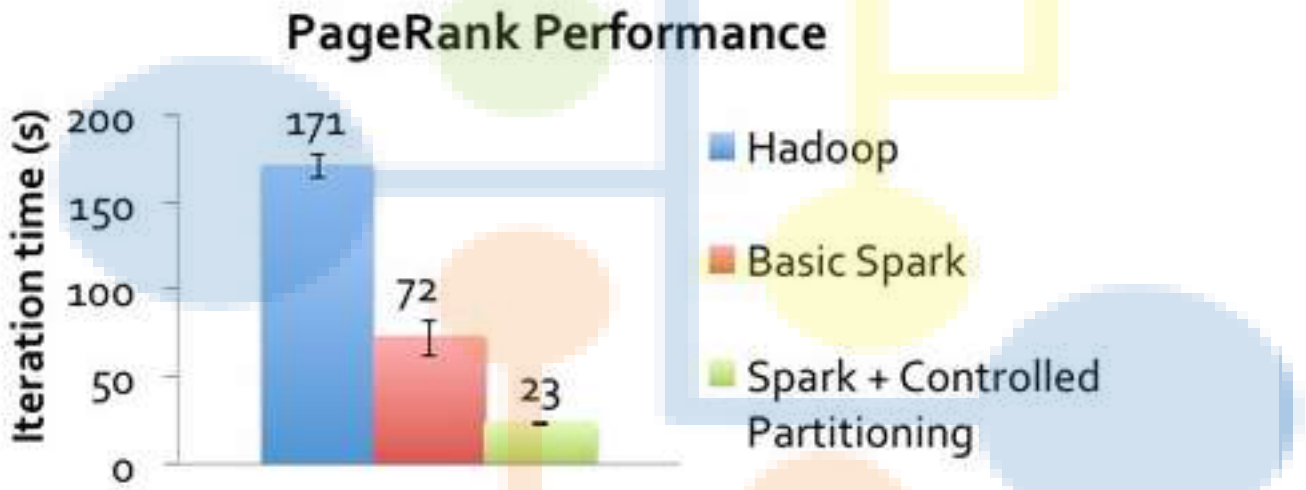
Computação na memória dos computadores em cluster!

Computação em Nuvem e em Cluster

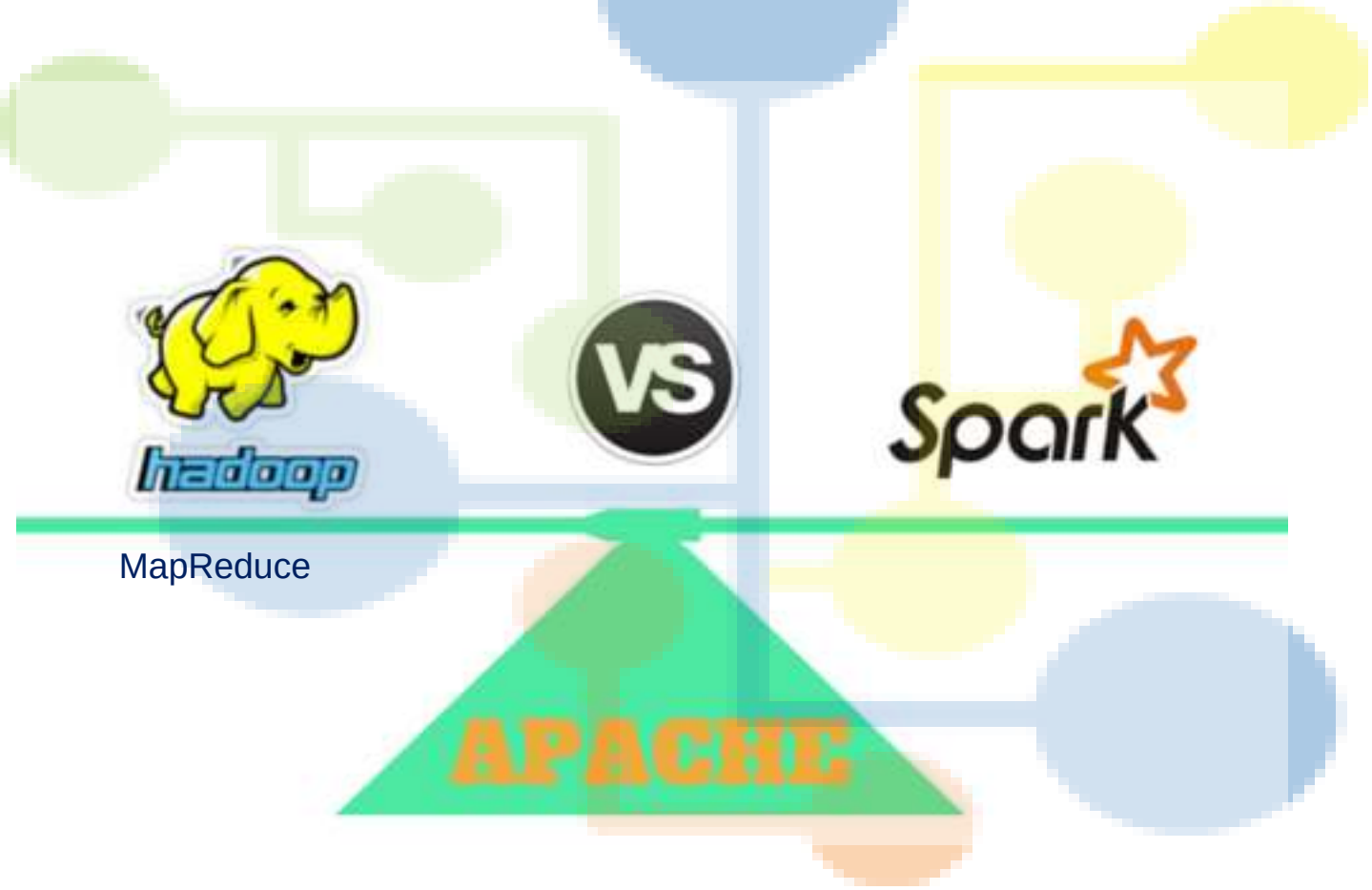


Computação em Nuvem e em Cluster



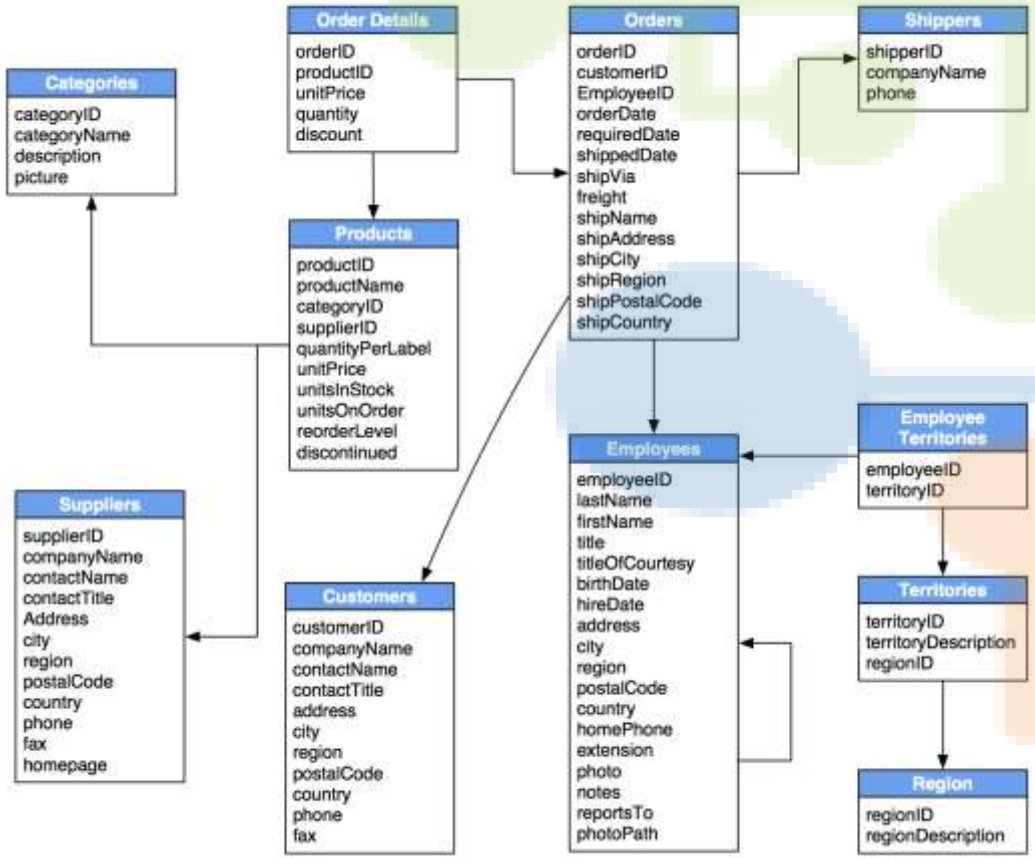

PageRank Performance
Iteration time (s)
200
150
100
50
0
171
72
23
Hadoop
Basic Spark
Spark + Controlled Partitioning

Computação em Nuvem e em Cluster



Linguagem SQL

Linguagem SQL

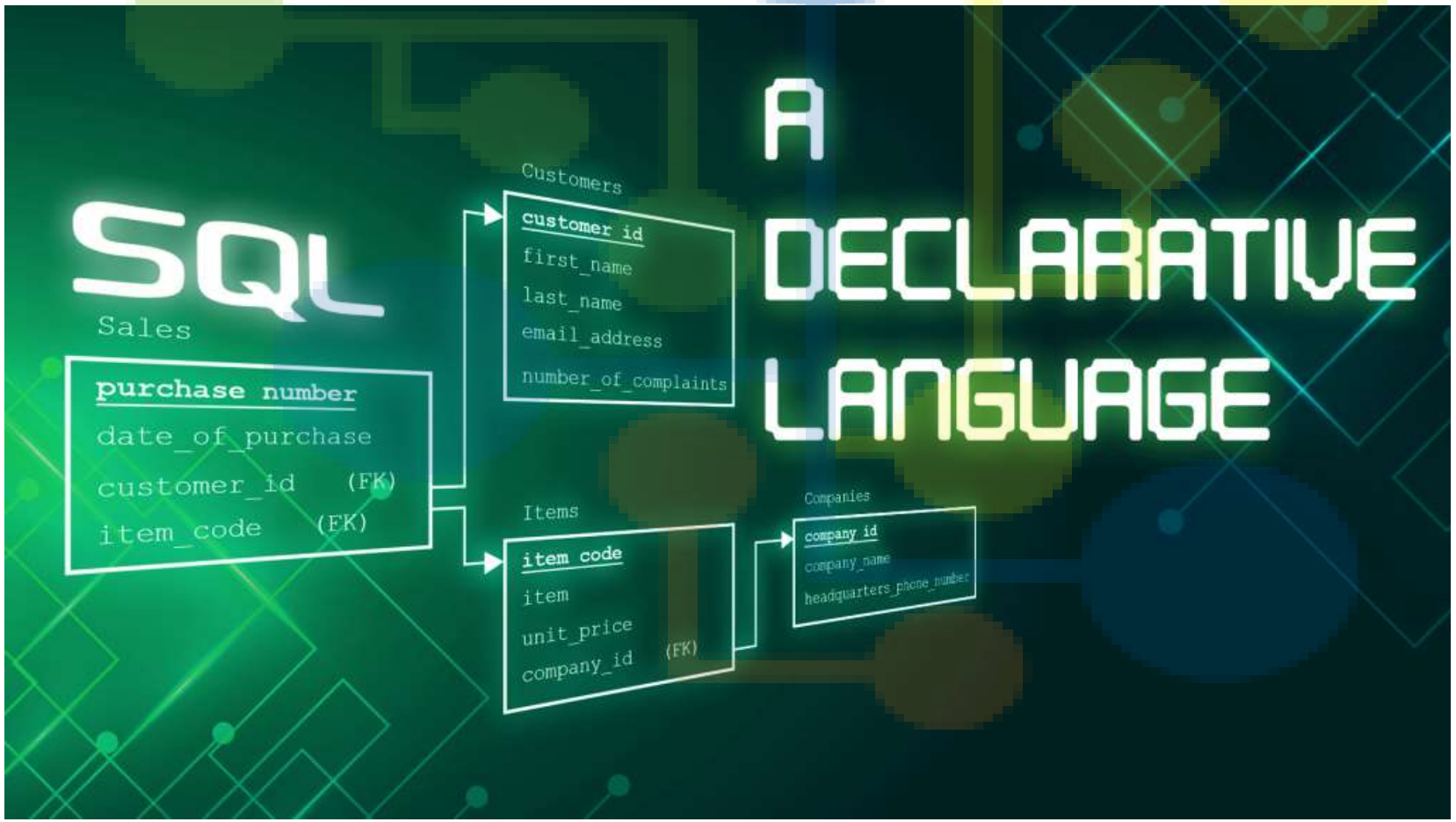


Usamos Linguagem SQL para consultar e manipular dados em bancos de dados.

Linguagem SQL

ANSI (American National Standards Institute)

Linguagem SQL



Linguagem SQL

Os comandos SQL se dividem em 3 tipos principais:

- **Linguagem de Manipulação de Dados ou DML (Data Manipulation Language)**
- **Linguagem de Definição de Dados ou DDL (Data Definition Language)**
- **Linguagem de Controle de Dados ou DCL (Data Control Language)**

Linguagem SQL

Linguagem de Consultas de Dados ou DQL (Data Query Language)

Linguagem SQL

Instruções SELECT:

- **Cláusulas** (FROM, WHERE, GROUP BY, HAVING, ORDER BY, DISTINCT)
- **Operadores Lógicos** (AND, OR, NOT)
- **Operadores de Comparação** (<, >, <>, <=, =, >=, BETWEEN, LIKE)
- **Funções de Soma** (AVG, COUNT, SUM, MIN, MAX)

Linguagem SQL

Instruções DML:

INSERT: utilizado para inserir registros em uma tabela.
Exemplo: INSERT into CLIENTE(ID, NOME) values(1000,'Obama');

UPDATE: utilizado para alterar valores de uma ou mais linhas de uma tabela.
Exemplo: UPDATE CLIENTE set NOME = 'Angela' WHERE ID = 1000;

DELETE: utilizado para excluir um ou mais registros de uma tabela.
Exemplo: DELETE FROM CLIENTE WHERE ID = 1000;

Linguagem SQL

Instruções DDL:

CREATE: utilizado para criar objetos no banco de dados.
Exemplo (criar uma tabela):
CREATE TABLE FORNECEDOR (ID INT PRIMARY KEY, NOME VARCHAR(50));

ALTER: utilizado para alterar a estrutura de um objeto.
Exemplo (adicionar uma coluna em uma tabela existente):
ALTER TABLE FORNECEDOR ADD TELEFONE CHAR(10);

DROP: utilizado para remover um objeto do banco de dados.
Exemplo (remover uma tabela):
DROP TABLE FORNECEDOR;

Instruções DCL:

GRANT: autoriza um usuário a executar alguma operação.
Exemplo (dar permissão de consulta na tabela fornecedor para o usuário Obama):

```
GRANT select ON fornecedor TO obama;
```

REVOKE: restringe ou remove a permissão de um usuário executar alguma operação.
Exemplo (não permitir que o usuário obama crie tabelas no banco de dados):

```
REVOKE CREATE TABLE FROM obama;
```

Linguagem SQL

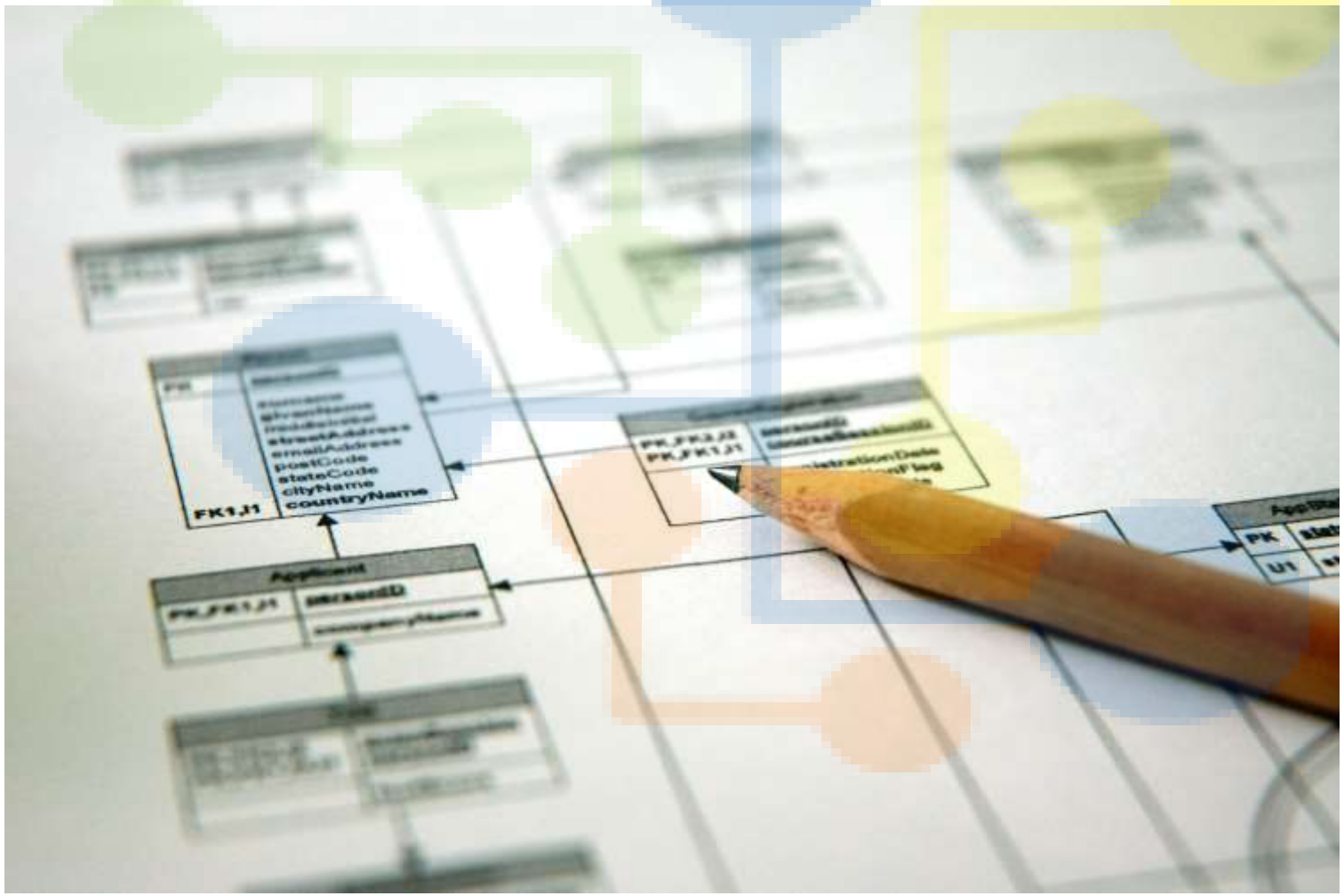
Instruções DTL (Linguagem de controle de transações)**:**

- BEGIN TRANSACTION
- COMMIT
- ROLLBACK

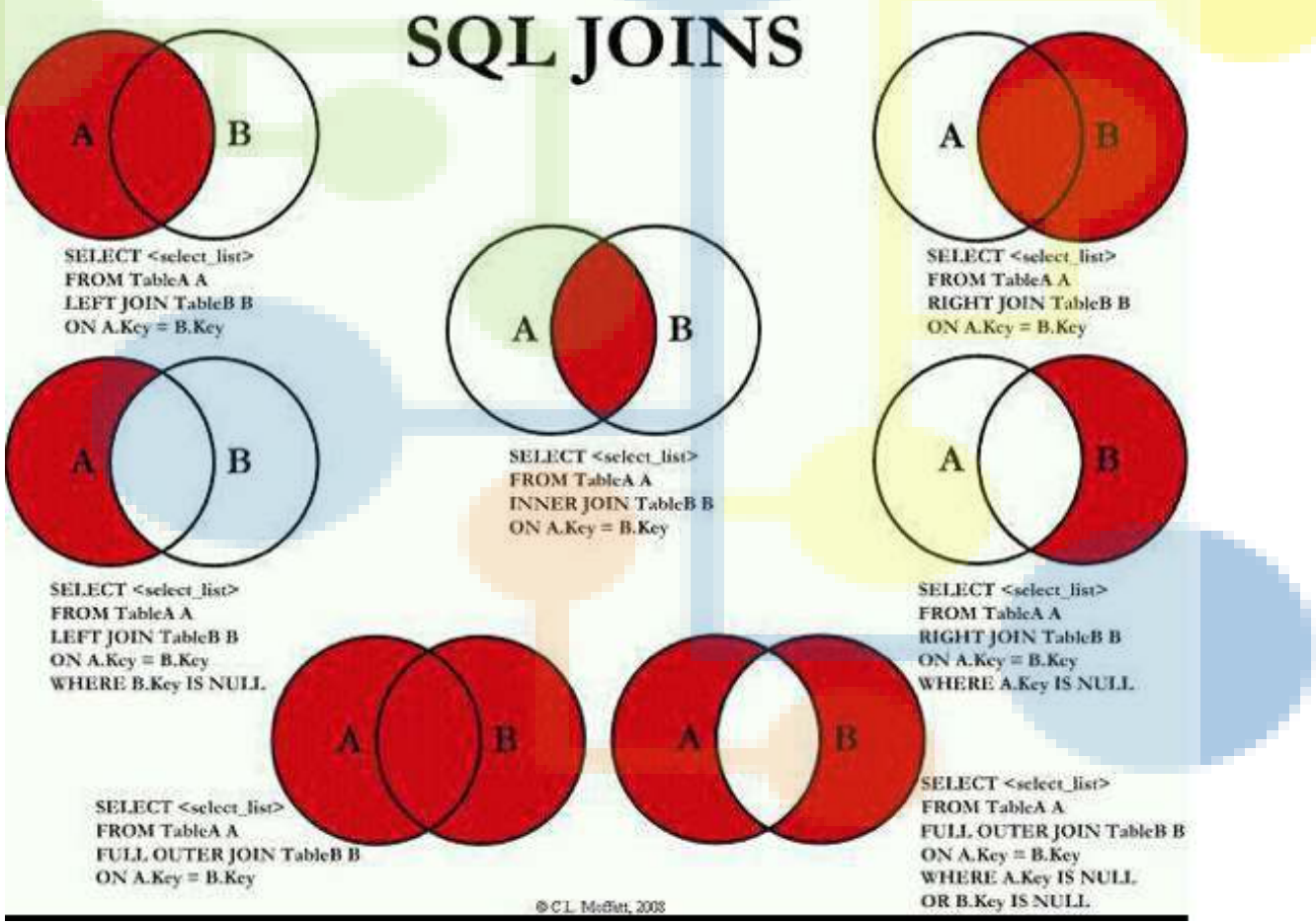
Terminologia

- Relação Relacionamento entre as tabelas
- Tabela
- Coluna (Atributo)
- Linhas (registros)
- Chave Primária (Primary Key)

Linguagem SQL



Linguagem SQL



Apache Spark SQL

Apache Spark SQL

Para dados estruturados

Apache Spark SQL

Principais Funcionalidades do Spark SQL:

API Data Source: com a inclusão de uma API para fontes de dados a biblioteca Spark SQL permite que a computação de informações sobre dados armazenados, de forma estruturada, seja facilitada e mais abrangente.
Servidor JDBC Interno: o Spark SQL traz um servidor JDBC (Java Database Connection) interno, permitindo a conexão a vários servidor de banco de dados

- Dataframes
- API Data Source
- Servidor JDBC Interno
- Funcionalidades para Data Science

Conexão com a MlLib

Componentes do Spark SQL

O Spark SQL possui 3 componentes principais:
DataFrame, Spark Session e SQL Context.

Apache Spark SQL

Mesmo conceito no Pandas, R e de tabelas: conjunto de dados estruturados. Podem ser transformados em RDD. Internamente, o Spark trata o Dataframe como RDD. A maior diferença entre dataframe e RDD é que no último se pode armazenar quaisquer tipos de dados, já o primeiro é para dados estruturados.

Dataframe

- RDDs já existentes
- Arquivos de dados estruturados
- Conjunto de dados JSON
- Tabelas Hive
- Banco de dados externos
- Mais eficientes que os RDD’s

Apache Spark SQL

Dataframe

Suporta as operações:

- filter
- join
- groupby
- agg
- Operações podem ser aninhadas

Apache Spark SQL

Spark Session

É um conector, uma espécie de API para acessar o Spark SQL.

- Criamos um Spark Session para acessar as funcionalidades do Spark SQL.
- Dataframes são criados a partir de Spark Sessions.
- Permite registrar um dataframe como uma tabela temporária e executar queries SQL sobre ele (muito útil no processamento de streams de dados).

Apache Spark SQL

SQL Context

Encapsula todas as funcionalidades relacionadas ao Spark. É possível criá-lo a partir do Spark Context (para acesso ao ambiente de cluster).

```
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
```

Fontes de Dados JDBC

Fornecida pelo Spark SQL

Isto é uma string de conexão

Tabelas Temporárias

Operações SQL podem ser executadas em tabelas temporárias que, embora sejam estruturas simples, são muito podersas. Uma query executada em uma tabela temporária retorna outro dataframe.

Apache Spark SQL

Uma das principais vantagens do Spark SQL é que você não precisa reaprender nada. Podemos usar os mesmos conceitos de SQL que usamos em bancos de dados relacionais, extrair dados para o Spark e então nos beneficiarmos do processamento paralelo e distribuído fornecido pelo framework Spark.

Tenha uma Excelente Jornada de Aprendizagem.

Muito Obrigado por Participar!

Equipe Data Science Academy