

# Tecnologias de Big Data

Os sistemas de *Data Warehouse* são tradicionalmente suportados por modelos multidimensionais predefinidos, tendo o intuito de prover suporte a aplicações de *Business Intelligence*. Tais modelos são geralmente implementados sobre bancos de dados relacionais e geridos por meio da linguagem SQL (*Structured Query Language*) (SALINAS e LEMUS, 2017). Embora SGBDs relacionais possam gerenciar grandes quantidades de informações, estes possuem certas limitações de escalabilidade horizontal, o que não ocorre com soluções de Big Data (MONGODB, 2022; SALINAS e LEMUS, 2017). A resposta às novas necessidades é a utilização de memória extensiva, distribuição de dados e paralelização de processamento, que de uma forma ou de outra estão incluídos no Apache Hadoop, Apache Spark, bases de dados NoSQL e tecnologias complementares a estas (SALINAS e LEMUS, 2017).

## Apache Hadoop

O Apache Hadoop é um framework de código aberto usado para armazenar e processar com eficiência grandes conjuntos de dados que variam em tamanho de gigabytes a petabytes. Em vez de usar um grande computador para armazenar e processar os dados, o Hadoop permite agrupar (*clustering*) vários computadores para analisar conjuntos de dados massivos em paralelo (AMAZON, 2022). O Apache Hadoop é constituído por quatro módulos:

- Hadoop Distributed File System (HDFS): sistema de arquivos distribuído que é executado em hardware padrão ou de baixo custo. Oferece melhor taxa de transferência de dados do que os sistemas de arquivos tradicionais, além de alta tolerância a falhas e suporte nativo de grandes conjuntos de dados.
- Yet Another Resource Negotiator (YARN): Gerencia e monitora clusters de nós e utilização de recursos. Permite o agendamento de trabalhos e tarefas.
- MapReduce: framework que ajuda os programas a fazer a computação paralela em dados. A tarefa do *map* pega os dados de entrada e os converte em um conjunto de dados que pode ser calculado em pares de valores-chave. A saída do *map* é consumida pelas tarefas de *reduce* para agregar a saída e fornecer o resultado desejado.
- Hadoop Common: Fornece bibliotecas Java comuns que podem ser usadas em todos os módulos.

## Apache Spark

O Apache Spark é um sistema de processamento distribuído de código aberto usado para processar cargas de trabalho de *big data*. Ele utiliza *in-memory* cache (cache em memória) e recursos de otimização de consultas para agilizar a execução de consultas analíticas em conjuntos de dados de qualquer tamanho. Ele fornece APIs de desenvolvimento em Java, Scala, Python e R e oferece suporte à reutilização de código em várias cargas de trabalho -- processamento em lote, consultas interativas, análise em tempo real, aprendizado de máquina e processamento de grafos. O Apache Spark é atualmente uma das estruturas de processamento distribuído de *big data* mais populares (AMAZON, 2022).

O Spark foi criado para resolver limitações do MapReduce, fazendo processamento em memória (in-memory), reduzindo o número de etapas de trabalho e reutilizando dados em várias operações paralelas. Com o Spark, apenas uma etapa é necessária onde os dados são lidos na memória, as operações são executadas e os resultados são gravados de volta, resultando em uma execução muito mais rápida.

O Spark também reutiliza dados usando um cache na memória para acelerar os algoritmos de aprendizado de máquina que chamam repetidamente uma função no mesmo conjunto de dados. A reutilização de dados é realizada por meio da criação de DataFrames, uma abstração sobre Resilient Distributed Dataset (RDD), que é uma coleção de objetos armazenados em cache na memória e reutilizados em várias operações do Spark. Isso reduz drasticamente a latência, tornando o Spark várias vezes mais rápido que o MapReduce, especialmente em trabalhos de aprendizado de máquina e análises interativas.

## Apache Spark vs. Apache Hadoop

Fora as diferenças no design do Spark e do Hadoop MapReduce, muitas organizações consideram essas estruturas de *big data* complementares.

O Hadoop é um framework open source que possui o Hadoop Distributed File System (HDFS) como armazenamento, o YARN como forma de gerenciar os recursos computacionais utilizados por diferentes aplicações e uma implementação do modelo de programação MapReduce como mecanismo de execução.

Spark é um framework de código aberto, focado em consultas interativas, aprendizado de máquina e cargas de trabalho em tempo real. Ele não possui seu próprio sistema de armazenamento, mas executa análises em outros sistemas de armazenamento como HDFS ou outras soluções populares como Amazon Redshift, Amazon S3, Couchbase, Cassandra e outras. O Spark sendo executado sobre o Hadoop aproveita o YARN para compartilhar clusters conjuntos de dados comuns, bem como outros recursos do Hadoop, garantindo níveis consistentes de serviço e resposta.

## Ecossistema Hadoop

O ecossistema Hadoop cresceu significativamente ao longo dos anos devido à sua extensibilidade. Hoje, o ecossistema Hadoop inclui muitas ferramentas e aplicativos para ajudar a coletar, armazenar, processar, analisar e gerenciar *big data*. Algumas das aplicações mais populares são (AMAZON, 2022):

**Spark** - Um sistema de processamento distribuído de código aberto comumente usado para cargas de trabalho de *big data*. O Apache Spark usa cache na memória e execução otimizada para desempenho rápido e oferece suporte a processamento geral em lote, análise de streaming, aprendizado de máquina, bancos de dados de grafos e consultas ad-hoc.

**Presto** - Um mecanismo de consulta SQL distribuído de código aberto otimizado para análise de dados ad-hoc de baixa latência. Ele suporta o padrão ANSI SQL, incluindo consultas complexas, agregações, junções e funções de janela. O Presto pode processar dados de várias fontes de dados, incluindo o Hadoop Distributed File System (HDFS) e o Amazon S3.

**Hive** - Permite que os usuários aproveitem o Hadoop MapReduce usando uma interface SQL, permitindo análises em grande escala, além de armazenamento de dados distribuído e tolerante a falhas.

**HBase** - Um banco de dados com versão não relacional e de código aberto executado no Amazon S3 (usando EMRFS) ou no Hadoop Distributed File System (HDFS). O HBase é um armazenamento de *big data* distribuído e massivamente escalável, criado para acesso aleatório, estritamente consistente e em tempo real para tabelas com bilhões de linhas e milhões de colunas.

## Bancos de Dados NoSQL

O termo NoSQL refere-se a um conjunto de sistemas de gerenciamento de bancos de dados baseados em estruturas não relacionais e sem um esquema pré-definido, que facilitam a escalabilidade horizontal e a gestão de dados não estruturados. Para melhorar o desempenho desses bancos de dados, as propriedades transacionais de conformidade ACID (*Atomicity, Consistency, Isolation, Durability*) não são garantidas, aderindo ao princípio de design BASE (*basic availability, soft state, eventually consistent*). O teorema CAP afirma que um sistema distribuído só pode fornecer simultaneamente duas propriedades entre consistência (C), disponibilidade (A) e tolerância à partição (P). A conformidade com ACID mantém consistência e disponibilidade, abrindo mão da possibilidade de implementações distribuídas em paralelo. O BASE, em vez disso, adota a tolerância à partição e, portanto, abre mão da disponibilidade imediata (CP) ou da consistência imediata (AP), dependendo do tipo de requisitos na camada de aplicação (SALINAS e LEMUS, 2017).

Bancos de dados NoSQL podem ser agrupados em bancos de dados documentais, de colunas (*column-family*), de valores-chave (*key-value*) ou grafos. Em bancos de dados orientados a documentos, cada registro é armazenado como um documento que é encapsulado e codificado em um formato padrão semi-estruturado como XML, JSON, BSON, etc.. Os bancos de dados orientados a documentos mais populares são MongoDB e CouchDB. Os bancos de dados orientados a colunas são caracterizados pela agregação de colunas de dados dentro de contêineres dados (*keyspace*). Algumas implementações deste tipo são Google BigTable, Cassandra, Apache HBase, Hypertable e Cloudata. Os bancos de dados orientados a valores-chave

são caracterizados por armazenar dados como pares de valores-chave. Isso funciona eficientemente na memória usando estruturas *map*, *arrays* associativos ou tabelas *hash*, e também gerencia o armazenamento persistente. Apache Accumulo, CouchDB, Amazon Dynamo e Redis, entre outros, são os bancos de dados de chave-valor mais usados. Por fim, os bancos de dados orientados a grafos são caracterizados por uma organização de armazenamento diferente, na qual cada nó representa um objeto que pode estar relacionado a outros objetos por meio de uma ou mais arestas direcionadas (que, também, podem ser objetos). Dentro desta categoria podemos citar Neo4J, AllegroGraph e FlockDB. Em certos casos, bancos de dados orientados a grafos podem ser configurados para cumprir as propriedades ACID (SALINAS e LEMUS, 2017).

Salinas e Lemus (2017) mencionam sobre a utilização de bancos de dados NoSQL para a construção de *Data Warehouses*. Considerando a utilização da Modelagem Dimensional na estruturação de *Data Warehouses*, aparentemente a utilização de bancos de dados NoSQL não aparenta ser a melhor opção, como demonstrado por Pereira, Oliveira e Rodrigues (2015). Em seu trabalho, estes autores comparam o desempenho de um DW construído sobre o SQL Server (banco de dados relacional) e o MongoDB (banco de dados NoSQL).

---

Revisão #5

Criado Wed, Mar 9, 2022 11:23 AM por FLAVIO LOPES DE MORAIS

Atualizado Wed, Dec 14, 2022 9:02 AM por FLAVIO LOPES DE MORAIS