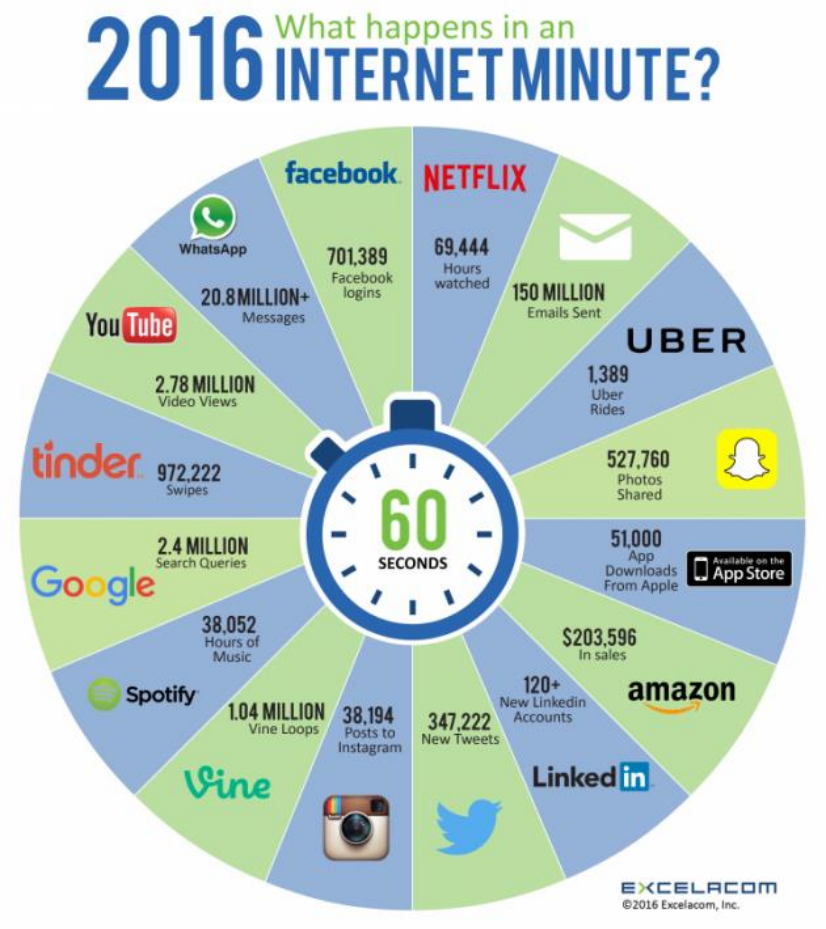
1 Introdução

Atualmente, os diversos dispositivos e sistemas têm gerado uma grande quantidade de dados. Se analisarmos os números de *sites* e mídias sociais como Youtube, Facebook, Twitter, Instagram, veremos que a cada dia o volume e a variedade de dados aumentam absurdamente. Para se ter uma noção desse universo, a imagem abaixo mostra o volume de dados que foram gerados a cada minuto na *internet* no ano de 2016.



Os bancos de dados e as aplicações de processamento tradicionais não conseguem processar esse grande volume de dados. Esse volume de dados fez com que grandes empresas buscassem alternativas mais baratas e melhores para conseguir processar esses dados.

2 O que é Big Data?

Uma das definições diz que Big Data é o conjunto de dados extremamente amplo que exige ferramentas especiais para processar esses dados em tempo hábil. De forma mais simples, é a análise de grandes conjuntos de dados para gerar valor para o negócio.

O termo ganhou força nos últimos anos devido a popularização dos *smartphones*, *smartwatches*, sensores, mídias sociais e serviços que tentam entregar para os usuários serviços mais adequados. O conceito ganhou força no começo dos anos 2000, quando o analista Doug Laney articulou a definição atualmente de big data em três Vs:

- **Volume**: está relacionado com o tamanho dos dados. Pode ser uma grande quantidade de arquivos pequenos ou mesmo grandes arquivos. Segundo estimativas, 1 minuto de vídeo no Youtube ocupa aproximadamente 45 MB de dados (https://www.youtube.com/watch?time_continue=136&v=B0DTW9ZLgZk)

- **Velocidade**: está relacionada à geração de dados. Organizações estão coletando ou gerando dados em volumes nunca antes imaginados. *Sites* de *e-commerce* geralmente armazenam cada *click* que um usuário faz no *site*; *smartphones* armazenam constantemente a localização dos usuários.

- **Variedade**: está relacionada ao formato dos dados. São gerados em inúmeros formatos – desde estruturados (geralmente, armazenados em tabelas de bancos de dados) a não-estruturados (documentos de texto, *e-mails,* vídeos, áudios).

  A variedade de dados é grande, e podem ser classificadas em três grupos:

- **Estruturados**: dados geralmente armazenados em tabelas de bancos de dados.

- **Semiestruturados**: dados que, apesar de não estarem em bancos de dados,

possuem marcações ou *tags* que dão a indicação semântica dos dados, geralmente, disponibilizados em formato xml ou json.

- **Não estruturados**: são arquivos diversos encontrados nos computadores, como pdf, doc, xls, txt, *e-mail*, imagem, vídeo, por exemplo. Esse tipo de dado não era muito explorado pelas soluções tecnológicas.

3 Introdução ao Hadoop

Até então, a forma de armazenamento e processamento exigia computadores com maior número de processadores e de memória, com sistemas redundantes como fontes e placas de rede, elevando os investimentos de TI das organizações. Além dos investimentos em *hardware*, também eram exigidas licenças de *software* igualmente caras. Para aumentar a capacidade de processamento desses ambientes, geralmente adicionavam-se mais processadores e mais memórias. O aumento da velocidade de leitura e escrita era feita trocando os discos por unidades mais rápidas. Esse modelo de expansão é conhecido como escalabilidade vertical.

Para reduzir o custo de montagem desse tipo de ambiente, empresas como o Google pensaram em uma solução que viabilizasse o grande poder de processamento com equipamentos mais baratos. Fazendo uso de computação distribuída, tem-se um conjunto de computadores dividindo o processamento e o armazenamento dos dados, e que, juntos, formam um cluster. Essa solução foi denominada Hadoop, que possui um conjunto de ferramentas responsáveis pelo processamento (conhecida como modelo MapReduce) e pelo armazenamento de grandes conjuntos de dados (Hadoop Distributed File Systems - HDFS).

O Hadoop é formado pela interligação de vários computadores de baixo custo, chamadas de *hardware commodities*. O HDFS, responsável pelo armazenamento, grava os dados em pelo menos três computadores, tornando um sistema tolerante a falhas. Como os dados são divididos em vários discos, tem-se uma alta velocidade de leitura e

gravação, isso sem necessitar de discos de alto desempenho,

Características do Hadoop?

- Capacidade de armazenar e processar grandes quantidades de qualquer tipo de dado, e rapidamente.
- Poder computacional. O modelo computacional distribuído do Hadoop permite processar grandes volumes de dados rapidamente. Quanto maior a quantidade de nós (computadores) no *cluster*, maior é o poder de processamento.
- Tolerância a falhas. Por padrão, o Hadoop possui replicação tripla de dados. Logo, se um nó do *cluster* falhar, o dado também está armazenado em pelo menos outros dois nós. Se um nó cai, os trabalhos são automaticamente redirecionados para outros nós para garantir que a computação distribuída não falhe. Isso também minimiza a necessidade de *backup* dos dados.
- Flexibilidade. Ao contrário dos bancos de dados relacionais tradicionais, você não precisa pré-processar os dados antes de armazená-los. Você pode armazenar seus dados o quanto quiser e decidir como usá-los depois. Isso inclui dados não-estruturados como texto, imagens e vídeos.
- Custo baixo. A estrutura *open-source* é gratuita e utiliza *hardwares* comuns para armazenar grandes quantidades de dados.
- Escalabilidade. Você pode aumentar facilmente o seu sistema para lidar com mais dados ao adicionar nós. Também é conhecida como escalabilidade horizontal.

O *cluster* é composto por 2 tipos de nós:

- NameNode: responsável por gerenciar onde cada arquivo está armazenado e controlar o acesso. Recomenda-se o *backup* dos dados do NameNode, pois em caso de perda, todos os dados do *cluster* são perdidos. Para minimizar as falhas, pode-se ter uma réplica, o Secondary DataNode.
- DataNode: responsável pela leitura e gravação dos arquivos, além de executar

operações de criação, exclusão e replicação de acordo com as instruções recebidas pelo NameNode. Como os dados são replicados, não necessitam de *backup* de dados.

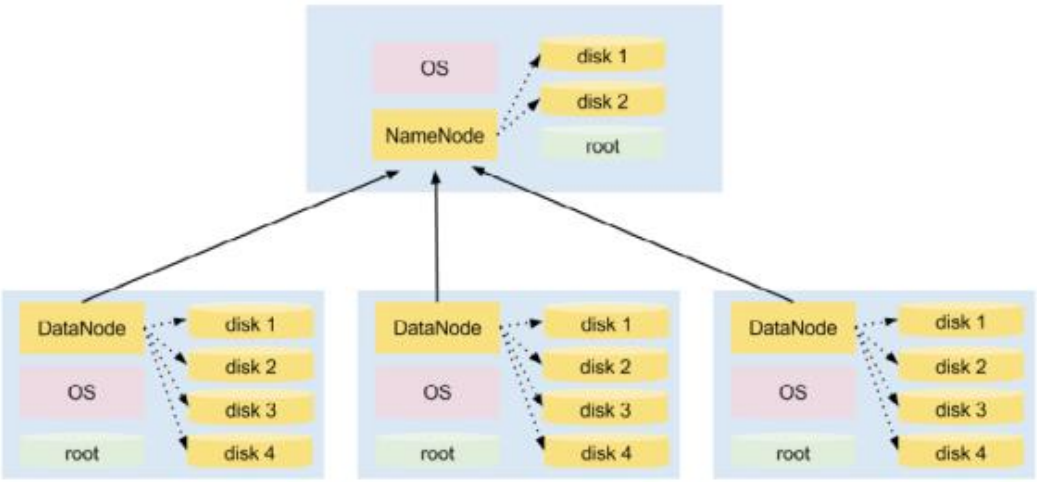
O Hadoop pode ser instalado gratuitamente por meio do *site* da Apache http://hadoop.apache.org, ou utilizando a distribuição de empresas conhecidas como Cloudera ou HortonWorks.

Para entender melhor o Hadoop, recomendo assistir as apresentações disponíveis em https://www.infoq.com/br/presentations/hadoop-bigdata/ e https://www.infoq.com/br/presentations/big-data-o-poder-da-informacao-seus-casos-de-uso-e-principais-arquiteturas/

4 Arquitetura Hadoop

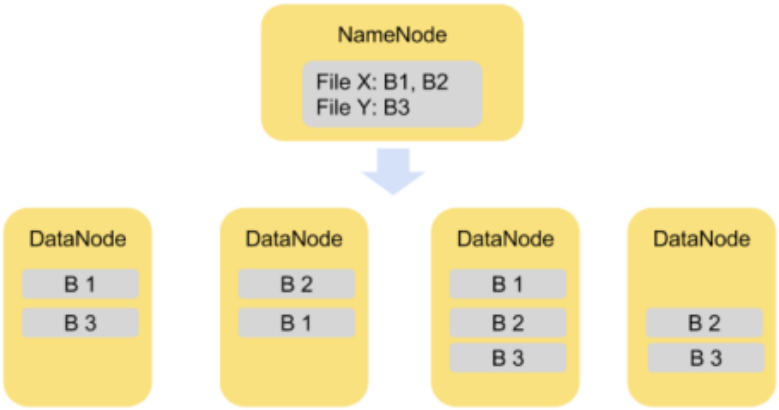
Na camada de armazenamento de dados há um sistema de arquivos distribuído, o Hadoop Distributed File System (conhecido apenas como HDFS), que fornece armazenamento escalável e tolerante a falhas. Ele foi projetado para escalar até centenas de *petabytes* e milhares de servidores. É otimizado para alto desempenho na leitura e escrita de grandes arquivos (acima dos *gigabytes*). O sistema se encarrega de dividir arquivos grandes em partes menores, normalmente blocos de 64MB, e replicar esses blocos pelos vários nós do *cluster* (geralmente três cópias), o que torna o processo tolerante a falhas, tanto em *hardware* quanto em *software*.

A imagem abaixo mostra o exemplo de um *cluster* Hadoop com quatro nós, um NameNode e três DataNodes. Ao gravar um arquivo no *cluster*, ele é divido em blocos e armazenado em três diferentes nós. Assim, se o um nó falhar, o dado estará replicado.



Fonte: https://dzone.com/storage/assets/9219790-dzone-refcard117-apachehadoop-new.pdf

Na figura abaixo temos o exemplo da gravação de dois arquivos: X e Y. O arquivo X é dividido em dois blocos: B1 e B2. O bloco B1 é armazenado nos nós 1, 2 e 3, já o bloco B2 é armazenado nos nós 2, 3 e 4. O arquivo Y ocupa apenas um bloco, B3, que é armazenado nos nós 1, 3 e 4. O NameNode armazena apenas a relação de blocos que formam um arquivo.



Fonte: https://dzone.com/storage/assets/9219790-dzone-refcard117-apachehadoop-new.pdf

Caso o nó 3 falhe, os blocos B1, B2 e B3 estarão presentes nos demais nós do

cluster.

Devido ao tamanho do bloco (64 MB) o HDFS não lida bem com o armazenamento de grandes quantidades de arquivos pequenos. Por exemplo, ao armazenar um arquivo com 1 MB, ele vai ocupar 64 MB. Além disso, para cada arquivo, o Hadoop aloca 150 *bits* de dados para gerenciar o arquivo, o que exige uma grande quantidade de memória. https://community.hortonworks.com/questions/167615/what-is-small-file-problem-in-hdfs.html

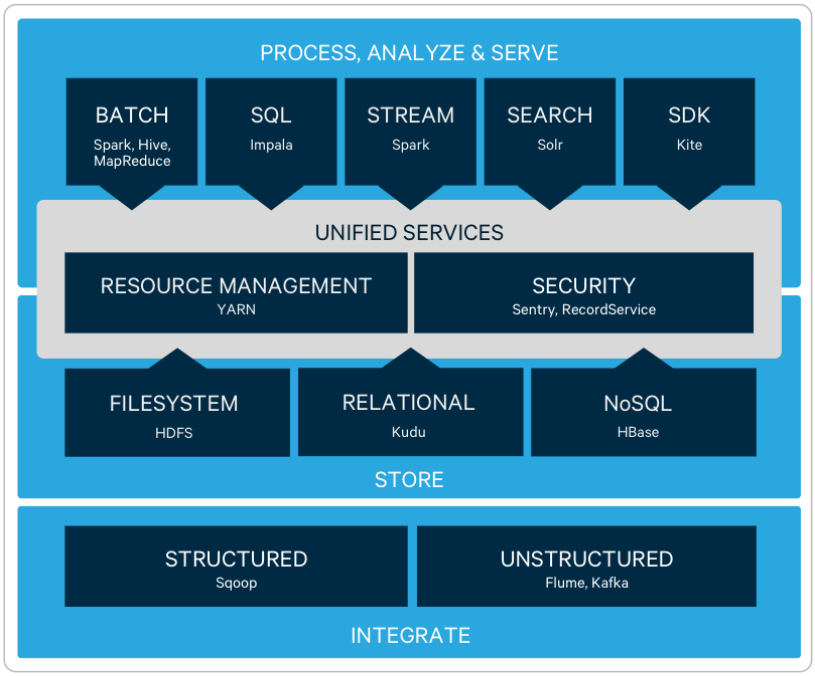
Na camada de processamento de dados há o MapReduce, utilizado para escrever aplicações massivamente paralelas que processam grandes quantidades de dados estruturados e não estruturados armazenados no HDFS. O MapReduce processa os dados onde estão armazenados, ou seja, em cada nó do *cluster*, diminuindo a quantidade de dados transmitida pela rede. Ele processa os blocos de arquivos nos nós onde estão armazenados.

Hadoop não é um banco de dados

Hadoop é um *framework* para armazenamento e processamento de grandes conjuntos de dados!

5 Ecossistema Hadoop

Com a popularização e evolução do Hadoop, novas ferramentas/componentes foram adicionadas ao ecossistema.



Fonte: https://www.cloudera.com/products/open-source/apache-hadoop.html

Na camada de integração ou ingestão de dados no Hadoop temos:

- **Sqoop**: utilizado para replicar dados de bancos de dados para o Hadoop ou vice-versa. Como os bancos de dados relacionais possuem limitações de capacidade e desempenho, principalmente para uso em aplicações analíticas, é comum a replicação dessas tabelas para um componente similar ao banco de dados utilizando o Scoop para essa atividade https://sqoop.apache.org/.
- **Flume**: utilizado para coletar, agregar e transferir grandes volumes de dados para armazenamento no HDFS. Muitas aplicações precisam coletar dados de diversas fontes, como *logs*, eventos, tabelas, agregar esses dados e encaminhar para um componente de armazenamento para ser usado em aplicações analíticas. https://flume.apache.org/
- **Kafka**: utilizado para criar *streaming* de dados de diversas fontes, principalmente em cenários onde análises RealTime são necessárias. Esse *streaming* de dados funciona como os sistemas de filas, onde existem diversos produtores de dados, que podem ser um sensor, um sistema ou outra coisa que encaminhe para o Kafka

se encarregar de entregar o dado para os interessados. Pode-se, por exemplo, automatizar o *streaming* de dados de uma tabela de banco para uma ferramenta de indexação e busca, ou seja, a cada dado novo no banco, ele é replicado em outra fonte de dados. https://kafka.apache.org/

Camada de serviços temos:

- **YARN**: é uma plataforma de gerenciamento de recursos computacionais em *cluster*, bem como pelo agendamento e monitoração de *jobs* de processamento. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html
- **Sentry**: é responsável pela segurança de acesso aos dados armazenados no *cluster* Hadoop. Chegando a ser 100 vezes mais rápido do que o MapReduce https://sentry.apache.org/

Camada de processamento e análise:

- **MapReduce**: modelo de processamento de grandes volumes em paralelo, que divide o trabalho em várias tarefas independentes executadas nos nós do *cluster*.
- **Spark**: é um *framework* para clusterização que executa o processamento em memória. https://spark.apache.org/
- **Hive**: fornece acesso aos dados armazenados no Hadoop traduzindo consultas SQL em processos de MapReduce. Facilitando a leitura, a escrita e o gerenciamento de grandes conjuntos de dados. Não é apropriado para aplicações transacionais que inserem ou atualizam registros frequentemente, é usado para grandes conjuntos de dados e atualizações. https://hive.apache.org/
- **Impala**: mecanismo nativo de acesso aos dados por meio de consultas SQL de alto desempenho. Muito utilizado por aplicações analíticas. https://impala.apache.org/

- Solr: ferramenta que provê indexação e busca textual escalável, tolerante a falhas e confiável. Pode ser usado para prover pesquisa similar ao Google entre dados que estão armazenados em banco de dados, arquivos não estruturados (pdf, csv, doc, xls, dentre outros) https://lucene.apache.org/solr/

Outros componentes do universo Big Data:

- Elasticsearch: suíte de componentes para armazenamento e indexação de dados que funciona em arquitetura distribuída, similar ao Solr. http://www.elastic.co

- Neo4J: banco de dados de grafo, muito útil para lidar com relacionamento de pessoas ou coisas. https://neo4j.com/ para conhecer o potencial de um banco de grafo recomendo assistir à apresentação https://www.infoq.com/br/presentations/quebrando-circulos-de-fraudes-em-cartoes-de-credito-com-neo4j

- CEPH: plataforma de armazenamento de objetos distribuída para armazenar objetos em equipamentos dos *clusters* https://www.redhat.com/pt-br/technologies/storage/ceph

Links úteis:

https://hadoop.apache.org/

https://br.hortonworks.com/

https://www.cloudera.com/

https://www.infoq.com/br/ai-ml-data-eng/

https://www.infoq.com/br/presentations/conhecendo-apache-hbase

https://www.infoq.com/br/presentations/utilizando-o-apache-kudu-como-workload-analitico/

https://www.devmedia.com.br/hadoop-mapreduce-introducao-a-big-data/30034

https://www.infoq.com/br/articles/apache-spark-introduction/

https://www.infoq.com/br/articles/mapreduce-vs-spark/

https://cio.com.br/hadoop-vs-spark-o-que-e-melhor-para-o-seu-negocio/

https://blog.mandic.com.br/artigos/ceph-armazenamento-em-bloco-do-seculo-xxi/

http://receita.economia.gov.br/orientacao/tributaria/cadastros/cadastro-nacional-de-pessoas-juridicas-cnpj/dados-publicos-cnpj