



GERÊNCIA DE INFRAESTRUTURA PARA BIG DATA

Tiago Coelho Ferreto – Aula 03

Pós-Graduação em
Ciência de Dados e Inteligência Artificial

Ementa da disciplina

Introdução à arquitetura para Big Data Analytics. Visão geral sobre Infraestrutura de armazenamento de dados para Big Data. Visão geral sobre Infraestrutura de computação e de rede para Big Data. Tópicos sobre virtualização e computação em nuvem. Plataformas de Big Data na nuvem: HDFS, Hadoop e MapReduce. Estudos de caso com Spark.

Professores

MARCOS TAKESHI

Professor Convidado

Especialista em Big Data na Semantix, que atua em diversos projetos de empresas do setor financeiro, telecom, varejo e saúde. Realiza análises de arquiteturas, infraestruturas, ambientes, sistemas e ferramentas big data, visando o correto funcionamento e performance. Formado em engenharia eletrônica pela Escola de Engenharia Mauá, pós-graduado em Administração de Empresas pela FGV-SP, MBA em Big Data na FIAP, e empreendedorismo no Babson College.

TIAGO COELHO FERRETO

Professor PUCRS

É professor adjunto da Pontifícia Universidade Católica do Rio Grande do Sul. Possui Doutorado em Ciência da Computação pela PUCRS (2010) com Doutorado sanduíche na Technische Universität Berlin, Alemanha (2007-2008). Tem experiência na área de Ciência da Computação, com ênfase em Redes de Computadores, atuando principalmente nos seguintes temas: computação em nuvem, grades computacionais, virtualização, processamento de alto desempenho e gerência de infraestrutura de TI.

Encontros e resumo da disciplina

AULA 1

Para ser um profissional de Data Science é necessário ter paciência e construir um bom Network.

Empresas tem grande interesse em processar os dados e deles extrair informação com a finalidade de monetizar.

É bom estar no meio de pessoas que saibam mais do que você, sempre você tem que estar no meio de pessoas melhores.

AULA 2

O Spark possibilita a obtenção de resultados imediatos.

É importante você saber e conseguir atuar em mais de uma frente.

Certificações podem mostrar que você tem conhecimento do assunto.

AULA 3

Nos últimos anos a gente tem, a cada ano, um novo software auxiliando no processamento de grandes volumes de dados.

Além de armazenar e processar, eu tenho que conseguir extrair valor.

O Hadoop como a principal ferramenta para trabalhar com grandes volumes de dados.

AULA 4

A redundância garante a persistência da informação.

O HDFS é a principal fonte de dados de entrada e saída do Hadoop.

Como utilizar as aplicações Sqoop e Flume.

AULA 5

MapReduce uma solução de escalonamento e capacidade de processamento.

Hadoop Streaming como implementação de funções Map e Reduce em linguagens diferentes de Java.

O Pig como linguagem alternativa para programar MapReduce.

AULA 6

O Hive trabalha com a linguagem SQL com interações através de linhas de comando em formato shell.

O Spark tem como benefícios uma melhor performance, extensibilidade e melhor suporte para outros cenários.

O componente principal do Spark é o RDD (Resilient Distributed Dataset).

MARCOS TAKESHI

Professor Convidado

MARCOS TAKESHI

Professor Convidado

TIAGO COELHO FERRETO **TIAGO COELHO FERRETO** **TIAGO COELHO FERRETO** **TIAGO COELHO FERRETO**

Professor PUCRS

Professor PUCRS

Professor PUCRS

Ciências de Dados e Inteligência Artificial

Gerência de Infraestrutura para Big Data

Prof. Tiago Ferreto

tiago.ferreto@pucrs.br

Quem sou eu?

- Tiago Ferreto
 - Professor da Escola Politécnica / PUCRS
 - Membro do PPGCC/PUCRS
 - Mestrado/Doutorado em Ciência da Computação
 - Líder do GRIN – Grupo de Pesquisa em Redes, Infraestrutura, e Computação em Nuvem
 - Coordenador do Laboratório de Alto Desempenho – IDEIA/PUCRS
- Principais interesses
 - Big Data
 - High Performance Computing
 - Cloud Computing/Edge Computing
 - IT Infrastructure
 - Networking



**PROGRAMA DE PÓS-
GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO / PUCRS**



Big Data nos velhos tempos...



1999

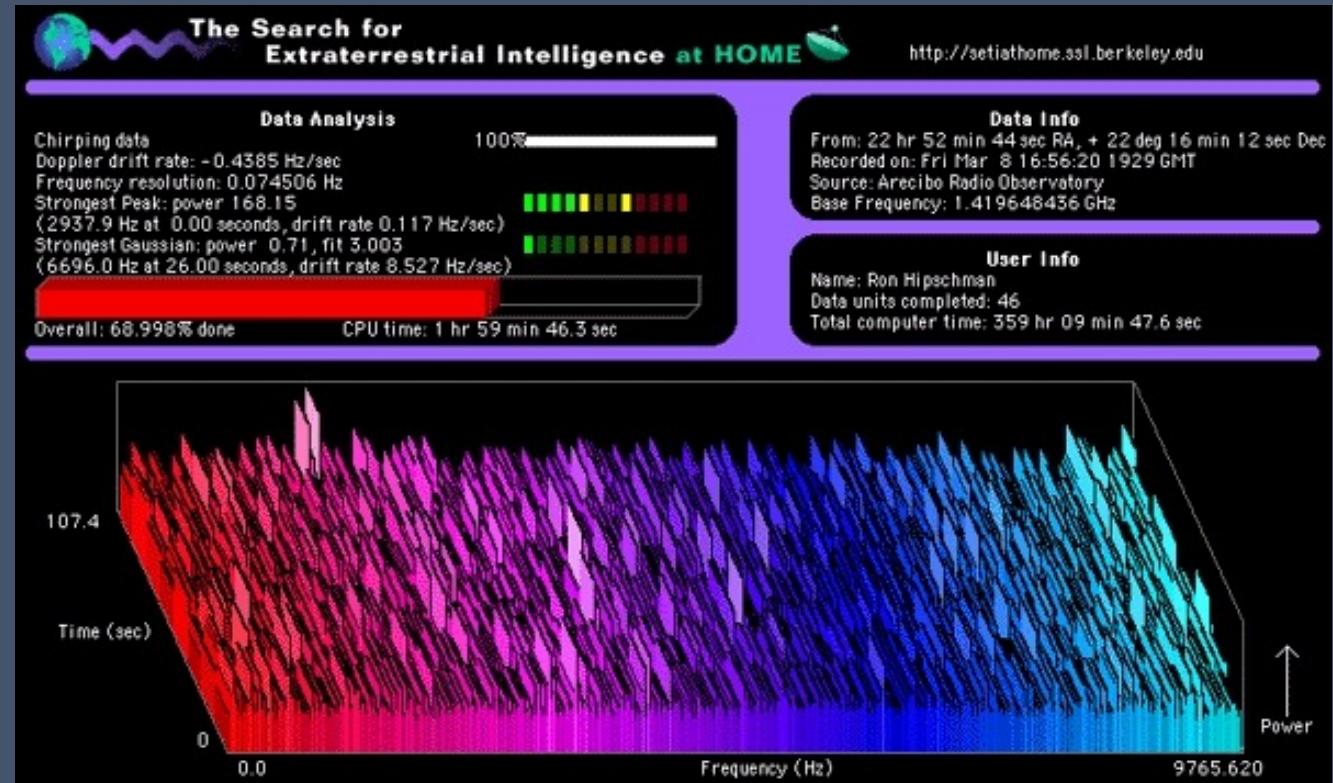


Observatório de Arecibo

2 TB = 2,5 dias de observação

Workpackage = 350 kB

screensaver



Grid Computing
Computação voluntária

Larga escala

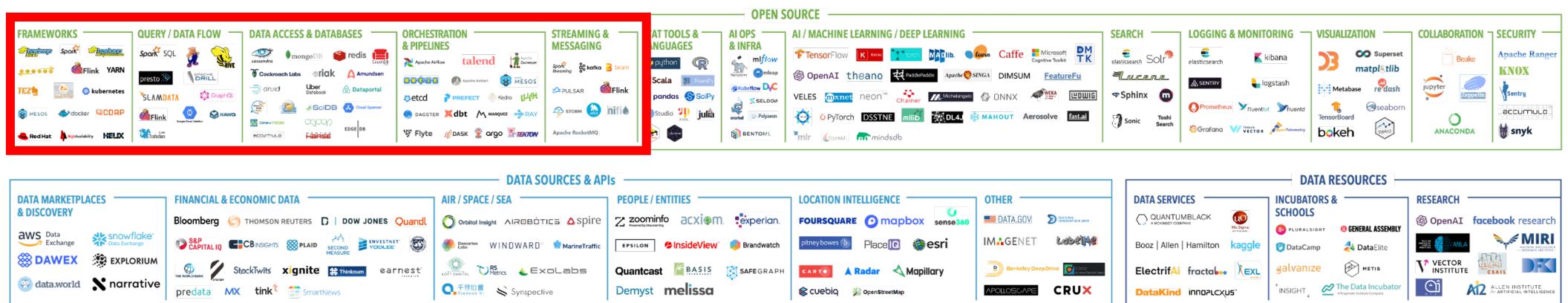
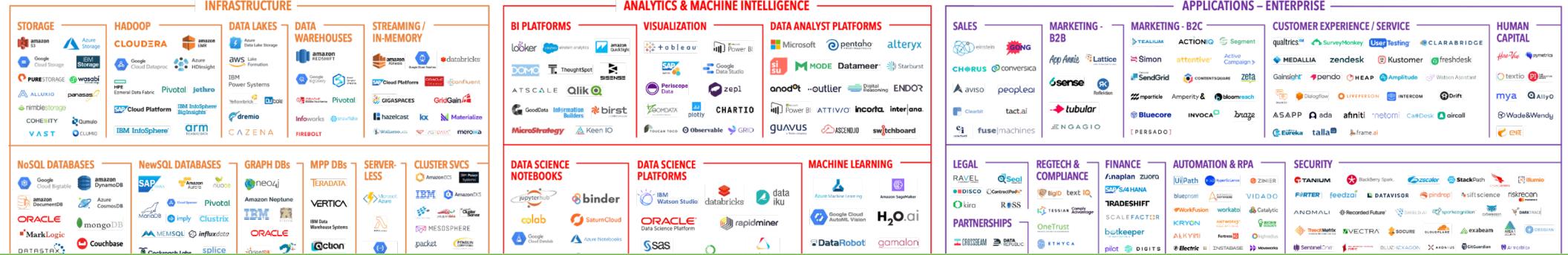
5,2 milhões de participantes

2 milhões de anos de tempo de computação agregado

Data center da Google - Douglas County, Georgia

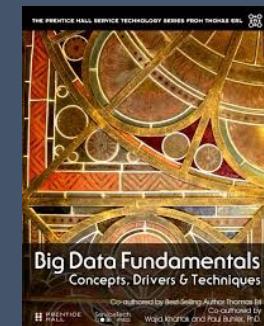
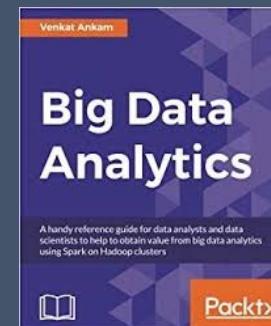
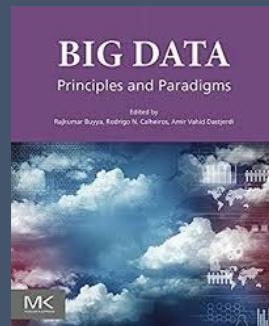
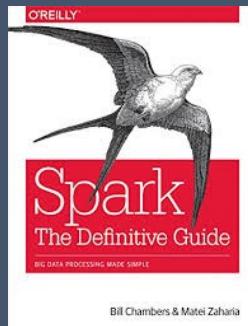
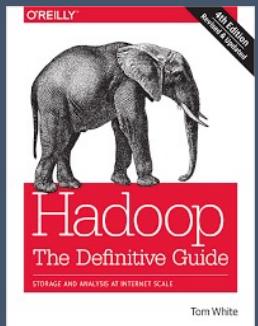


Google Datacenters - <https://www.google.com/intl/pt-BR/about/datacenters/>



Referências

1. RAJ, Pethuru (Ed.). Handbook of research on cloud infrastructures for big data analytics. IGI Global, 2014.
2. GUPTA, Sumit et al. Real-Time Big Data Analytics. Packt Publishing Ltd, 2016.
3. RYZA, S.; LASERSON, U.; OWEN, S.; WILLS, J. Advanced Analytics with Spark: Patterns for Learning from Data at Scale. O Reilly Media, 2017.



Atividades práticas

- Pacotes do ecossistema Hadoop
- Docker
- Jupyter

<https://github.com/tiagoferreto/HadoopJupyter>

Ciências de Dados e Inteligência Artificial

Gerência de Infraestrutura para Big Data

Prof. Tiago Ferreto

tiago.ferreto@pucrs.br

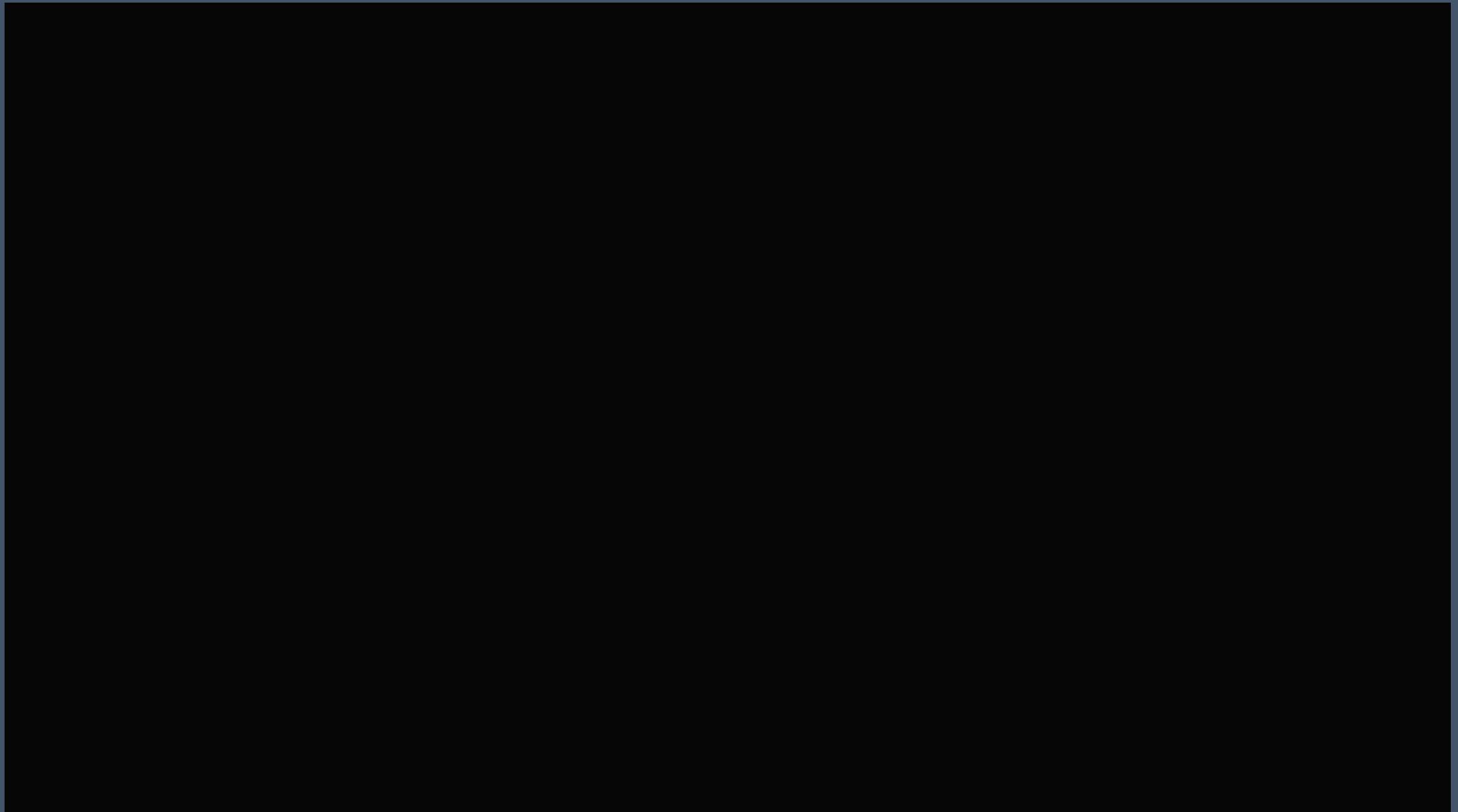
Ciências de Dados e Inteligência Artificial

Gerência de Infraestrutura para
Big Data

BIG DATA

Prof. Tiago Ferreto

tiago.ferreto@pucrs.br



Era dos dados!

- Estimativa da IDC (*International Data Corporation*) do tamanho do “universo digital”
 - 4.4 zettabytes em 2013 → **44 zettabytes em 2020**
- **Dilúvio de dados**
 - Bolsa de New York Stock gera aproximadamente 4–5 terabytes de dados por dia
 - Facebook hospeda mais de 240 bilhões de fotos, com um aumento de 7 petabytes por mês
 - Internet Archive armazena aproximadamente 18.5 petabytes de dados
 - Large Hadron Collider, produz cerca de 30 petabytes de dados por ano
- **Máquinas (IoT) irão gerar mais dados do que pessoas**
 - Logs, leitores RFID, redes de sensores, GPS, transações de varejo

Mais importante do que armazenar dados é extrair o valor deles!

Big Data

“Dados de grande volume, em relação ao Sistema de processamento, com uma variedade de dados estruturados e não estruturados contendo diferentes padrões de dados a serem analisados.”

Akhtar, Syed Muhammad Fahad. *Big Data Architect's Handbook: A guide to building proficiency in tools and systems used by leading big data experts.* Packt Publishing Ltd, 2018.

- Fatos
 - Abordagens tradicionais para armazenamento e processamento NÃO funcionam devido ao grande volume de dados
 - Os dados podem conter informações valiosas → precisam ser processados em um curto espaço de tempo
 - Essas informações valiosas podem ser usadas para fazer análises preditivas, bem como para marketing e muitos outros fins
 - Usando a abordagem tradicional, a análise não será concluída dentro do prazo determinado



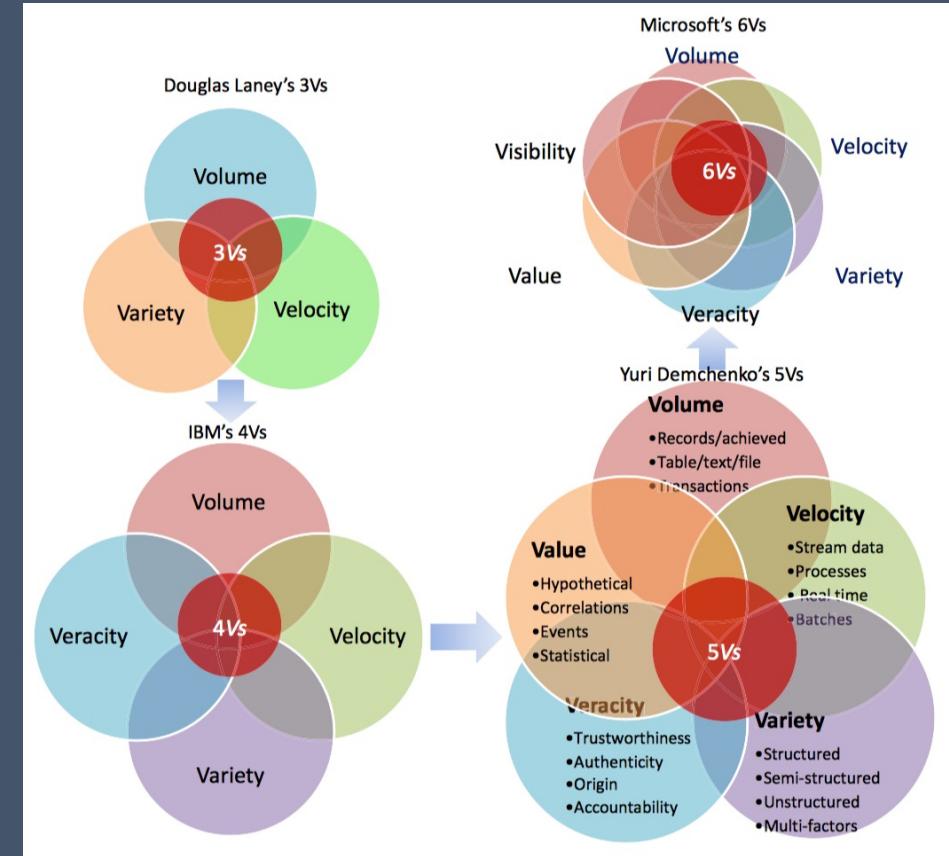
Fonte: <https://www.visualcapitalist.com/every-minute-internet-2020/>

Como medir?

1 Bit	Dígito binário
8 Bits	1 byte
1,024 Bytes	1 KB (kilobyte)
1,024 KB	1 MB (megabyte)
1,024 MB	1 GB (gigabyte)
1,024 GB	1 TB (terabyte)
1,024 TB	1 PB (petabyte)
1,024 PB	1 EB (exabyte)
1,024 EB	1 ZB (zettabyte)
1,024 ZB	1 YB (yottabyte)
1,024 YB	1 brontobyte
1,024 brontobyte	1 geopbyte

Dimensões de Big Data

- 3Vs (interpretação Gartner - publicado em um artigo por Douglas Laney em 2001)
 - Volume: escala de dados (entrada e cumulativa)
 - Velocidade: análise de dados de streaming
 - Variedade: diferentes formas de dados
- 4Vs (definição da IBM)
 - Volume, Velocidade, Variedade
 - + **Veracidade: incerteza dos dados**
- 6Vs (Microsoft)
 - Volume, Velocidade, Variedade, Veracidade
 - + **Variabilidade: complexidade do dataset (ausência de padrões)**
 - + **Visibilidade: requer uma visão completa dos dados para tomar decisões**
- 5Vs (Yuri Demchenko, 2014)
 - Volume, Velocidade, Variedade, Veracidade
 - + **Valor: importância dos resultados obtidos através do armazenamento, processamento e análise**



Volume

- Passado: as empresas usavam apenas dados criados por seus funcionários
- Agora: dados são gerados por dispositivos (IoT) e clientes
- A mídia social aumenta a quantidade de dados gerados (vídeos, fotos, Tweets, etc)
 - 6 bilhões (de 7 bilhões - população mundial) têm telefones celulares
 - Os telefones celulares possuem sensores que geram dados para cada evento, que são armazenados e analisados (exemplo, relatórios de saúde)
- O volume em big data é de uma quantidade que não pode ser coletada, armazenada e processada usando abordagens tradicionais
- Exemplos
 - Facebook
 - 2 bilhões de usuários ativos
 - 600 TB de dados são inseridos diariamente no banco de dados do Facebook
 - Avião a jato
 - 10 TB de dados são gerados para cada hora de voo (escala de petabytes gerados por dia para todos os voos)

Velocidade

- Taxa em que os dados são gerados - a rapidez com que os dados chegam
- Exemplos
 - A bolsa de valores de Nova York captura 1 TB de dados durante cada sessão de negociação
 - 120 horas de vídeo são enviadas ao YouTube a cada minuto
 - Os carros modernos têm quase 100 sensores para monitorar cada item, desde o combustível e a pressão dos pneus até os obstáculos ao redor
 - 200 milhões de e-mails são enviados a cada minuto
- Outra dimensão da velocidade é o período de tempo durante o qual os dados farão sentido e serão valiosos
 - Ele envelhecerá e perderá valor com o tempo ou terá um valor permanente?

Variedade

- Classificação de dados: estruturados ou não estruturados
- **Dados estruturados**
 - Informações com esquema predefinido ou modelo de dados (colunas predefinidas, tipos de dados, etc)
 - Exemplos: bancos de dados relacionais
- **Dados não estruturados**
 - Nenhum esquema ou modelo de dados predefinido
 - Exemplos: documentos, e-mails, mensagens de texto de mídia social, vídeos, imagens estáticas, áudio, gráficos, saída de sensores, dispositivos, etiquetas RFID, registros de máquinas, sinais de GPS de telefones celulares, etc.
- Variedade de dados - exemplos de dados não estruturados gerados
 - 400 milhões de Tweets são enviados por dia
 - 4 bilhões de horas de vídeos são assistidos no YouTube todos os meses
- Independentemente de sua “estrutura” - os dados devem ser processados a fim de gerar uma melhor experiência do usuário ou receita para as próprias empresas

Veracidade

- Incerteza sobre os dados
- Pode ser devido à baixa qualidade dos dados ou ruído nos dados
- A veracidade é importante ao analisar e tomar decisões
- Em certas situações, não há tempo para limpar dados de streaming ou dados de alta velocidade para eliminar a incerteza
 - Os dados podem perder valor se demorarem muito para serem analisados (por exemplo, dados gerados por sensores)
 - A incerteza deve ser levada em consideração

Variabilidade

- **Falta de consistência ou padrões fixos nos dados**
- Diferente da variedade
- Se o significado e a compreensão dos dados continuarem mudando, isso terá um grande impacto em sua análise e nas tentativas de identificar padrões

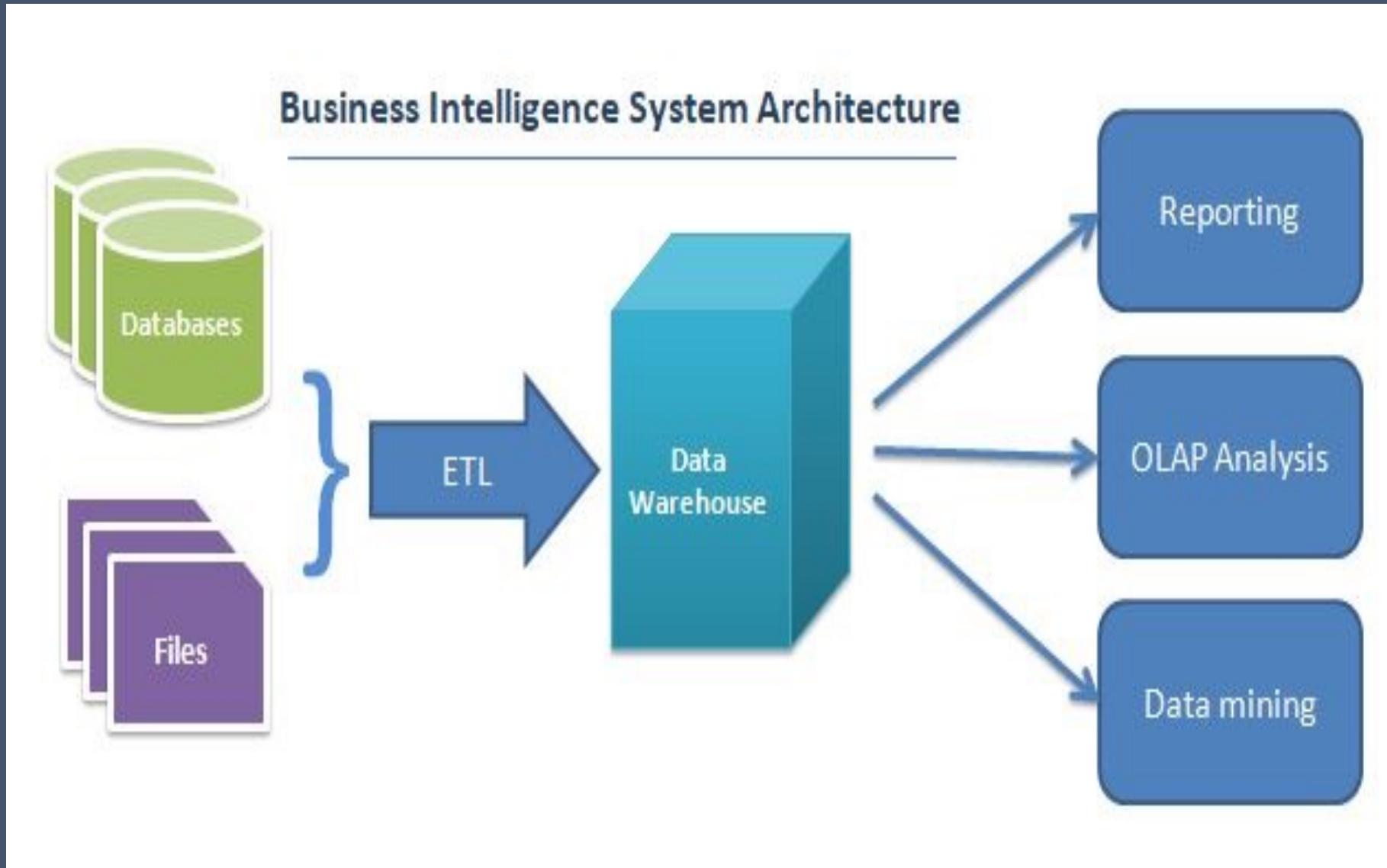
Valor

- Característica mais importante!
- Também se aplica a dados pequenos
- Avalia se vale a pena armazenar os dados e investir em infraestrutura para fazer isso
- Um aspecto do valor é que é necessário armazenar uma grande quantidade de dados antes que possam ser usados para fornecer informações valiosas
- Não há ganho se os dados armazenados não retornam em valor para a organização

Abordagens tradicionais para armazenamento de dados

- Os **trabalhos em lote** são programados para migrar dados para *Data Warehouses* em um período de dia, semana ou mês
- Os dados têm um **esquema** e são categorizados como **dados estruturados**
- Eles passam por um **ciclo de análise** para **criar conjuntos de dados e extrair informações significativas**
- Ingestão no Data Warehouse são realizadas por meio de operações ETL (*Extract, Transform, and Load*) - pega dados brutos e processa-os para relatórios e análise
- Data Warehouse é otimizado somente para relatórios e análise dos dados
- Sistema principal de Business Intelligence (BI)

Arquitetura de um Sistema de BI tradicional



Problemas com a abordagem tradicional

- **Latência**

- Os relatórios não são em tempo real (levam dias ou semanas para serem gerados)
- As organizações precisam responder rapidamente

- **Fontes limitadas**

- Depende de dados estruturados.
- Atualmente a maioria das informações não é estruturada (postagens de blogs, análises da web, fontes de mídia social, postagens, Tweets, fotos, vídeos, áudio, registros de sensores, etc.)

- **Escala limitada**

- Não conseguem lidar com uma quantidade crescente de dados gerados rapidamente

Cluster Computing

- Conjunto de computadores (nós) conectados atuando como uma única máquina para o usuário final
- Características
 - **Alta disponibilidade** - os dados devem estar disponíveis o tempo todo, mesmo quando ocorrerem falhas de hardware ou software
 - **Pool de recursos** - vários computadores são conectados entre si e os seus recursos (CPU, memória, rede, disco) são compartilhados para processar diferentes tarefas de forma independente
 - **Fácil escalabilidade** - permite adicionar capacidade de armazenamento ou poder computacional incluindo novas máquinas ao grupo (scale out ao invés de scale up)
 - **Processamento paralelo** - cada nó executa uma tarefa em paralelo
 - **Hardware de commodity** - usa hardware COTS (componentes de prateleira) com custo mais barato do que um servidor especializado

Nuvem versus infraestrutura local (on-premises)

	On-premises	Nuvem
Custo	Grande custo inicial. Precisa configurar servidores de alta tecnologia, rede, armazenamento.	Evita os custos de inicialização. Aumente sob demanda e pague por uso.
Segurança	Sensação de maior segurança. Controle sobre quem está acessando seus dados, quando são usados e com que finalidade.	Não há garantias sobre onde estão os dados, como estão sendo gerenciados ou quem pode acessá-los. Os provedores visam garantir que cada bit de informação colocado na nuvem seja seguro e protegido (por exemplo, mecanismos de criptografia, redundância e replicação geográfica).
Capacidades	Requer pessoal local para gerenciar a implementação no local - para configuração, suporte.	Também requer pessoal, mas com foco no problema.
Escalabilidade	Requer avaliação adicional da capacidade necessária para adquirir hardware. Dependendo da análise de demanda de capacidade, pode haver uma grande quantidade de recursos ociosos.	Os recursos podem ser adicionados sob demanda

Conjuntos de dados públicos

- AWS Public Dataset Program - <https://aws.amazon.com/opendata/public-datasets/>
- Google Cloud Public Datasets - <https://cloud.google.com/public-datasets/>
- Registry of Research Data Repositories - <https://www.re3data.org/>
- Wikipedia downloadable dataset –
http://en.wikipedia.org/wiki/Wikipedia:Database_download
- Million Song Dataset - <https://labrosa.ee.columbia.edu/millionsong/>
- U.S. Government's open data - <https://www.data.gov/>
- Open Data Brazilian Portal - <http://dados.gov.br/dataset>
<https://github.com/awesomedata/awesome-public-datasets>
- Kaggle Datasets - <https://www.kaggle.com/datasets>
- DataHub - <https://datahub.io/>
- NYC Taxi & Limousine Commission Trip Record Data -
http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
- YFCC100M - <https://multimediacommons.wordpress.com/yfcc100m-core-dataset/>

Exemplos de aplicações que usam Big Data

- Astrometry.net (<http://astrometry.net/>)
 - Observa o grupo de Astrometria no Flickr para novas fotos do céu noturno
 - Analisa cada imagem e identifica de qual parte do céu ela é, bem como quaisquer corpos celestes interessantes, como estrelas ou galáxias
- Google Books Ngram Viewer (<https://books.google.com/ngrams/>)
 - Usa o conjunto de dados do Google Livros para exibir um gráfico que mostra como as frases ocorreram em um corpus de livros (por exemplo, "British English", "English Fiction", "French") durante os anos selecionados.
- OpenFlights.org (<https://openflights.org/>)
 - Mapeia voos ao redor do mundo, pesquise e filtre de todas as maneiras, calcula estatísticas automaticamente e compartilha voos e viagens.

Ciências de Dados e Inteligência Artificial

Gerência de Infraestrutura para
Big Data

BIG DATA

Prof. Tiago Ferreto

tiago.ferreto@pucrs.br

Ciências de Dados e Inteligência Artificial

Gerência de Infraestrutura para
Big Data

HADOOP

Prof. Tiago Ferreto

tiago.ferreto@pucrs.br

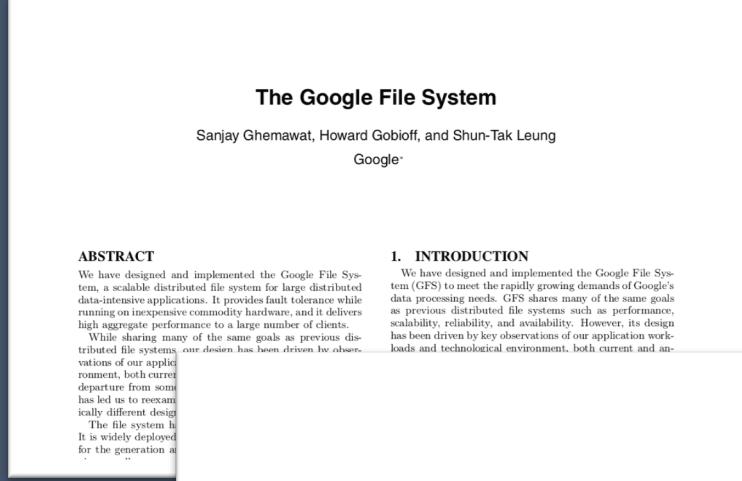
Big Data e História do Hadoop

- Início dos anos 2000
 - Surgimento de **metodologias de armazenamento e processamento** (“Big Data”) de **provedores de mecanismos de busca** (Google e Yahoo!)
- Provedores de mecanismos de busca
 - Primeiro grupo a enfrentar problemas de escala da Internet

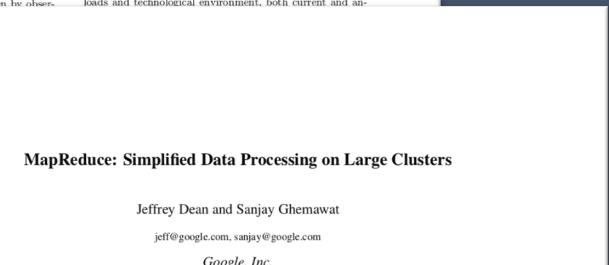
“Como processar e armazenar índices de todos os documentos na Internet?”

Big Data e História do Hadoop

- Linha do tempo
 - 2003 - Google lança o artigo **“The Google File System”** - sistema de arquivos escalável e tolerante a falhas
 - 2004 - Google lança **“MapReduce: Simplified Data Processing on Large Clusters”** - processamento em grande escala
 - Ao mesmo tempo - Doug Cutting começa a trabalhar no **Nutch** (projeto de web crawler)
 - Inspirado pelos documentos do Google, ele decide incorporar os princípios de armazenamento e processamento, que mais tarde são movidos para um novo projeto chamado Hadoop
 - 2006 - **Hadoop** nasce como um projeto de código aberto sob a Apache Software Foundation



The Google File System
Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google



MapReduce: Simplified Data Processing on Large Clusters
Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com
Google, Inc.



Nutch
Abstract
MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.
Programs written in this functional style are automatically parallelized by the system.

Hadoop
Abstract
given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.
As a reaction to this complexity, we designed a new abstraction that allows us to express the simple computa-

Big Data e História do Hadoop

- Ao mesmo tempo em que o projeto Hadoop foi criado, outras inovações tecnológicas estavam acontecendo (**dilúvio de dados**)
 - Expansão rápida do **comércio eletrônico**
 - Nascimento e rápido crescimento da **Internet móvel**
 - **Blogs** e conteúdo da web direcionado ao usuário
 - **Mídia social**
- Além do Hadoop, também levou ao surgimento de outros projetos, como **Spark**, **Messaging Systems**, **NoSQL** , etc

O que é Hadoop?

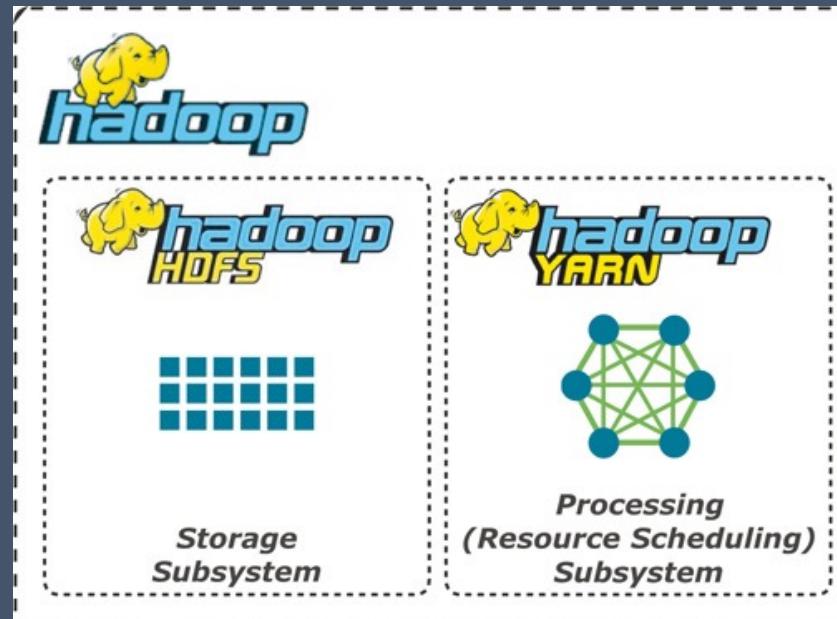
- Hadoop é uma **plataforma de armazenamento e processamento de dados**
- Big data torna muito difícil mover dados
- Conceito central do Hadoop → **LOCALIDADE DE DADOS**
 - A localidade dos dados refere-se ao processamento dos dados onde residem
 - A computação é enviada para os dados, em vez da computação solicitar dados de sua localização (por exemplo, DBMS)
- Hadoop permite o processamento de grandes conjuntos de dados localmente nos nós de um cluster
 - Usa uma **abordagem de nada compartilhado (*shared nothing*)**
 - Cada nó processa independentemente um subconjunto de todo o conjunto de dados sem precisar se comunicar um com o outro

O que é Hadoop?

- Sistema de *schema* durante a leitura (*schema-on-read*) → não existe esquema para operações de gravação
 - Pode armazenar e processar uma ampla variedade de dados, como:
 - Documentos de texto não estruturados
 - Documentos semiestruturados JSON ou XML
 - Extratos bem estruturados de sistemas de banco de dados relacionais
- Em sistemas *schema-on-write*, como bancos de dados relacionais, os dados são fortemente tipados e um esquema é predefinido e aplicado em todas as operações
 - Os bancos de dados NoSQL são sistemas *schema-on-read*
- Prós: não precisa de índices, estatísticas ou construções de otimização
- Contras: conjuntos de dados com tamanho maior e dados replicados
- O Hadoop usa uma **abordagem de dividir para conquistar** para resolver problemas aplicando conceitos de localidade de dados e nada compartilhado (*shared nothing*)

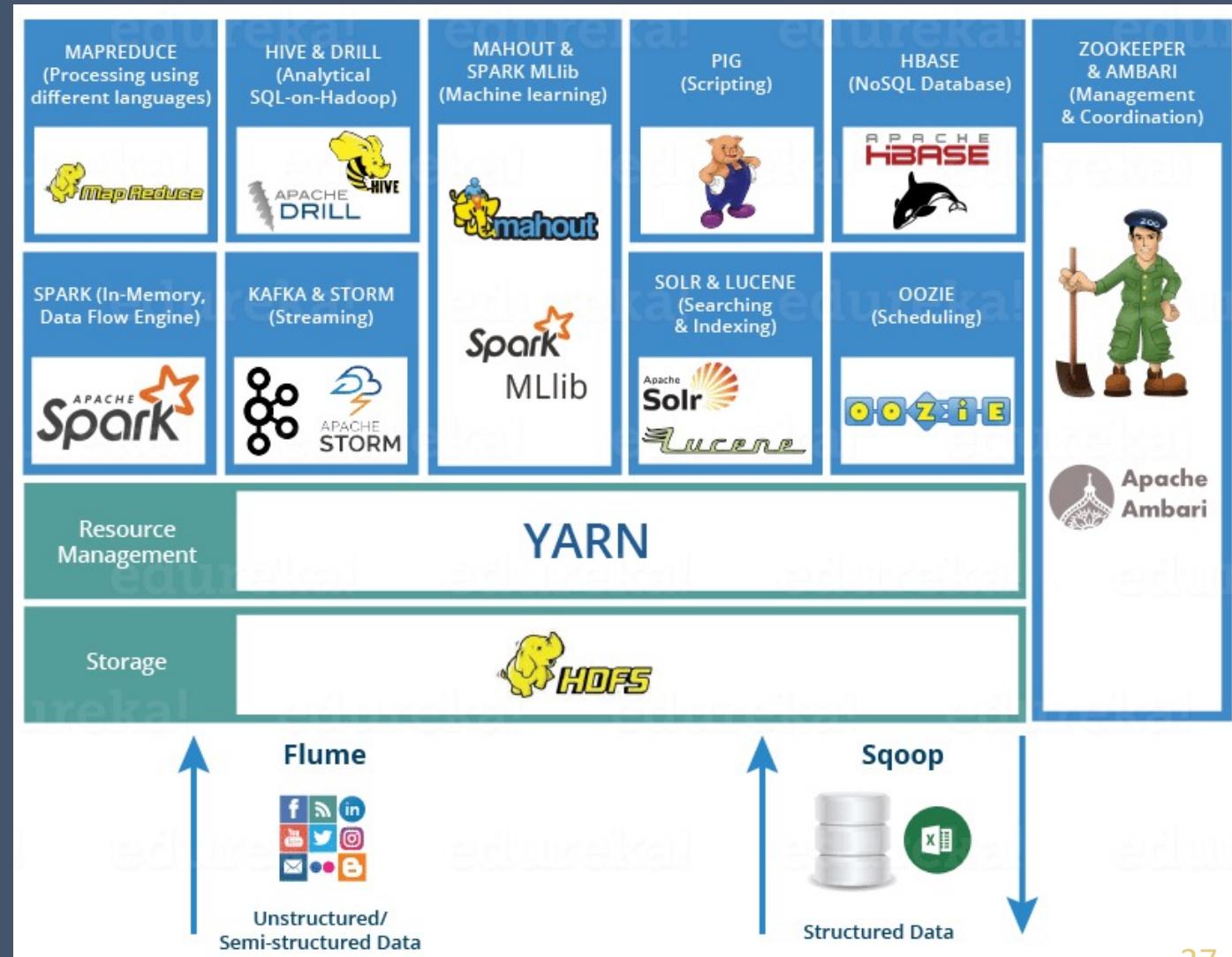
Hadoop – Componentes

- **Hadoop Common** - bibliotecas e utilitários necessários para outros módulos
- **HDFS (Hadoop Distributed File System)** - sistema de arquivos distribuído para armazenamento de dados
- **Hadoop YARN (Yet Another Resource Negotiator)** - plataforma para gerenciamento de recursos de computação e escalonamento de aplicações
- **Hadoop MapReduce** - implementação do modelo de programação MapReduce



Componentes do ecossistema Hadoop

- O ecossistema Hadoop inclui outros projetos que interagem ou se integram com o Hadoop
 - Ingestão de dados
 - Análise de dados
 - Bancos de dados NoSQL
 - Segurança
 - Gestão
 - Pesquisa
 - Aprendizado de Máquina
 - Coordenação de fluxo de trabalho
 - etc



Cenário comercial do Hadoop

- 2008 – **Cloudera**
 - CDH (Cloudera Distribution of Hadoop)
- 2009 – **MapR**
 - Inclui MapRFS - solução de armazenamento personalizada, adaptada de HDFS
- 2011 - **HortonWorks**
 - HDP (Hortonworks Data Platform)
- Cloudera, Hortonworks e MapR são considerados fornecedores de Hadoop “pure play”
 - Modelos de negócios são baseados no Hadoop
 - Existem outros fornecedores com distribuições proprietárias (o negócio principal não é Hadoop) - IBM, Pivotal, Teradata
- 2014 - ODPi (Open Data Platform Initiative) fundada por Hortonworks - odpi.org
 - Objetivo: fornecer um conjunto consistente e comercial do núcleo do Hadoop e componentes do ecossistema selecionados
 - Inclui outros fornecedores: IBM, Teradata, EMC, SAS, Pivotal, Splunk e outros
- 2018 - **Fusão entre HortonWorks e Cloudera** (concluído em janeiro de 2019)
- 2019 - HPE adquire MapR

Releases do Hadoop

- Hadoop 1 (2012)
 - Primeira versão - HDFS e MapReduce
- Hadoop 2 (2013)
 - Apresenta o YARN - otimiza a alocação de recursos para várias aplicações
- Hadoop 3 (2017)
 - Suporta vários NameNodes no HDFS (resolve o problema de SPOF)
 - HDFS usa *erasure coding* - reduz a sobrecarga de armazenamento
 - Utilização de containers (Docker)
 - Suporta a utilização de GPUs para processamento (Deep Learning)

Casos de uso típicos do Hadoop

- Descarregamento de data warehouse
 - Armazenamento de longo prazo e rotinas ETL
 - Aproveita os **recursos de armazenamento e distribuição de processamento de baixo custo** do Hadoop (em comparação com plataformas especializadas de Data Warehouse)
 - **Implementação de um Data Lake** - armazenamento de dados não estruturados e não processados antes de serem integrados em um Data Warehouse ou expostos em um Data mart

Casos de uso típicos do Hadoop

- **Armazenamento de eventos e processamento de eventos complexos**
 - Associado à IoT (Internet of Things) - milhares de endpoints, incluindo sensores de temperatura, RFID, scanners NFC, câmeras de CFTV, sistemas de sinalização, etc.
 - **Ingestão e processamento de fontes de dados de streaming**, como dados de sensores, mensagens ou logs
 - O Hadoop fornece **armazenamento de baixo custo para acumular milhões de mensagens e uma plataforma computacional** para extrair volumes de mensagens para identificar padrões ou peculiaridades
 - Normalmente **envolve outras plataformas de ecossistema Hadoop**, como Storm, Flume, Spark Streaming, Kafka ou outros

Casos de uso típicos do Hadoop

- **Análise Avançada (Ciência de Dados)**
 - Processo de **identificação de padrões em dados e construção de modelos matemáticos para representar esses padrões**, que podem ser usados para prever resultados de dados subsequentes
 - A análise avançada abrange as disciplinas específicas de:
 - Mineração de dados, modelagem preditiva e aprendizado de máquina (normalmente abordagens de resolução de problemas baseadas em pesquisa, iterativas e baseadas em tentativa e erro)

O Hadoop permite análises avançadas em larga escala!

Visão geral da arquitetura do cluster Hadoop

- Cluster YARN – Processamento
- Cluster HDFS – Armazenamento
- Embora ambos os clusters possam ser usados independentemente, um Hadoop Cluster é quando ambos os clusters (HDFS e YARN) são configurados nos mesmos nós

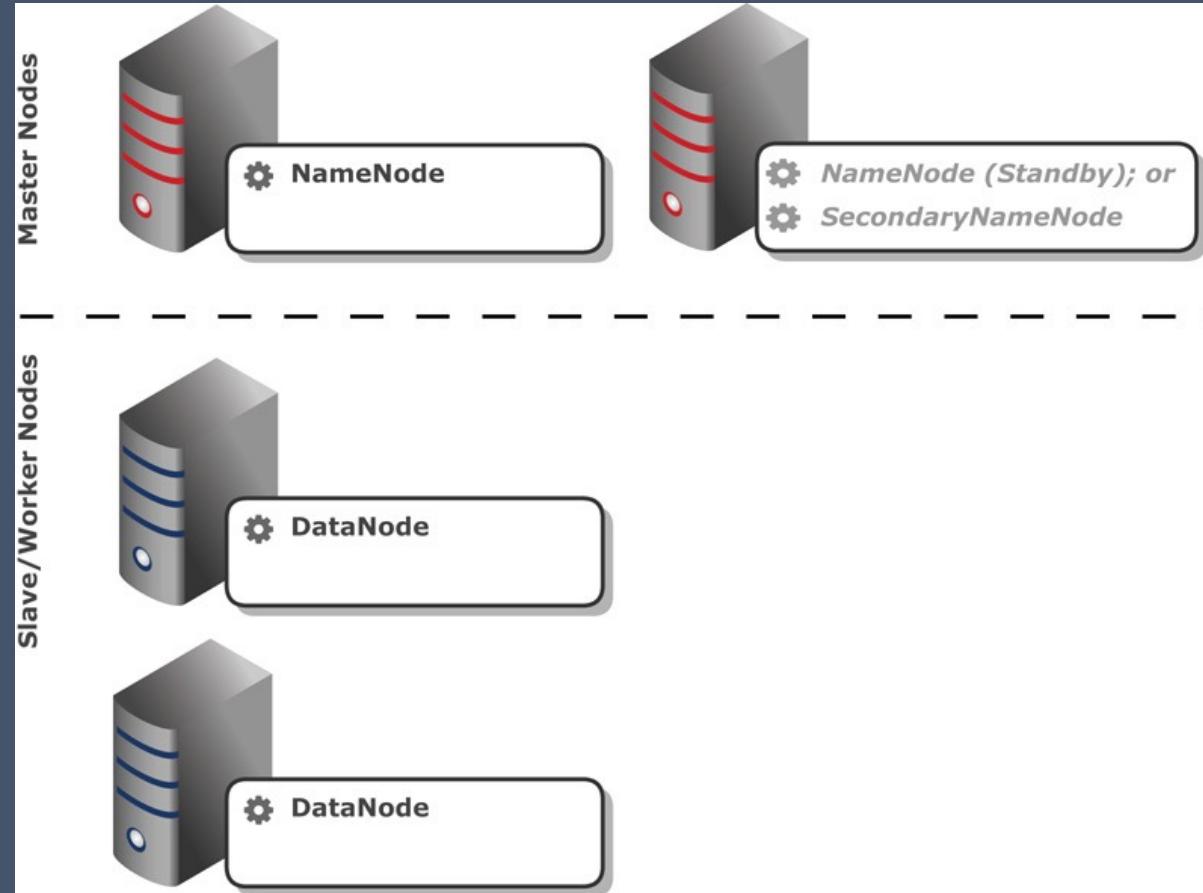
Conceitos básicos do HDFS

- Conjunto de **processos Java** para gerenciar e orquestrar o sistema de arquivos distribuído do Hadoop
- **Arquitetura de cluster mestre-escravo**
 - Modelo de comunicação em que **um processo tem controle sobre um ou mais processos**
 - No HDFS, **os processos mestre e escravo são pré-designados** (funções estáticas)
 - **Os processos HDFS são executados como daemons / serviços em nós do cluster e são classificados como mestre ou escravos**

Conceitos básicos do HDFS

- Arquivos, blocos e metadados
 - Os arquivos são compostos por blocos (normalmente 64 ou 128 MB) no HDFS
 - Os blocos são distribuídos e replicados em um ou mais nós do cluster
 - Os metadados do sistema de arquivos contêm informações sobre diretórios virtuais, arquivos e blocos físicos que compõem os arquivos
- **NameNode**
 - Processo do nó mestre HDFS
 - Responsável por gerenciar os metadados do sistema de arquivos
- **DataNode**
 - Processo do nó escravo HDFS
 - Responsável por gerenciar o armazenamento em bloco e acesso para leitura / gravação de dados
- Secondary NameNode and Standby NameNode
 - Processos opcionais

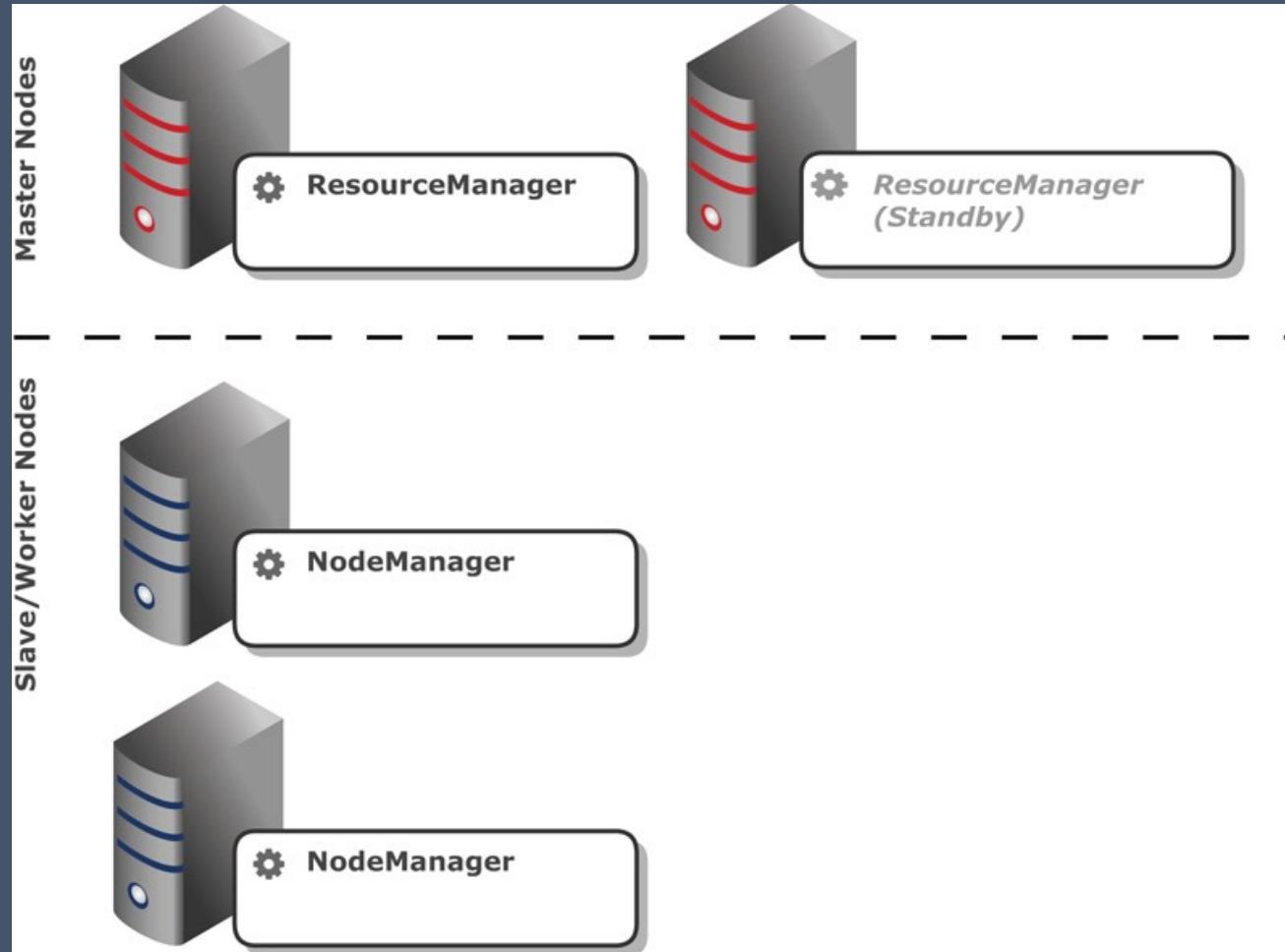
Visão geral do cluster HDFS



Conceitos básicos do YARN

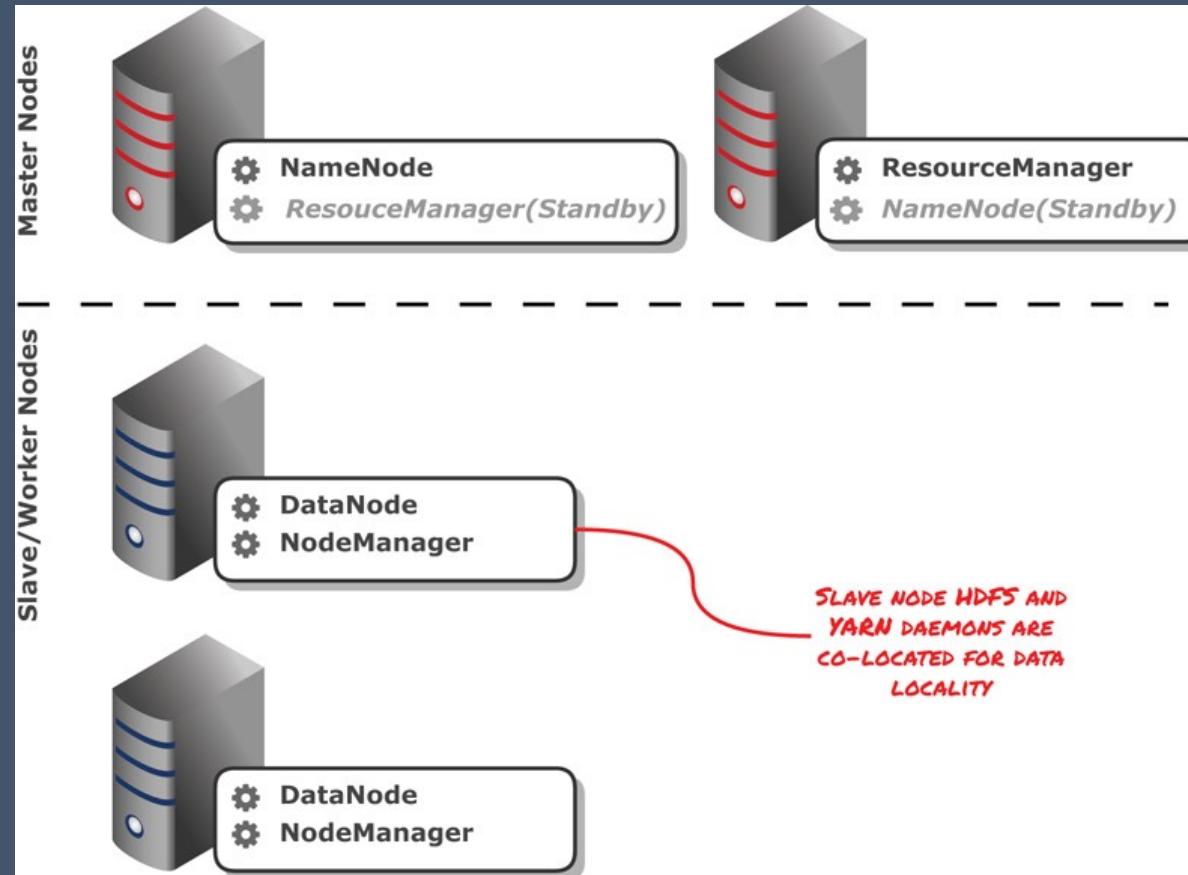
- Usa um modelo mestre-escravo como no HDFS
 - ResourceManager e NodeManager
- **ResourceManager**
 - Processo mestre do YARN
 - Responsável por **escalonar aplicativos e conceder recursos de computação de cluster (containers)**
- **NodeManager**
 - Daemon de nó escravo
 - **Gerencia containers no host**
- **ApplicationMaster**
 - Primeiro container alocado pelo ResourceManager para uma aplicação
 - Responsável pela aplicação
 - **Solicita os recursos necessários para o RM**
 - **Orquestra a alocação de recursos para executar as etapas da aplicação**
 - **Monitora o progresso das tarefas, estágios e dependências**

Visão geral do cluster YARN



Arquitetura do cluster Hadoop

- Combina os clusters HDFS e YARN



Modos de implantação do cluster Hadoop

- Hadoop oferece suporte a três modos de implantação
 - **LocalJobRunner** (operação *standalone*)
 - Executa uma aplicação localmente em uma única JVM (Java Virtual Machine)
 - Permite a integração da aplicação com um IDE (por exemplo, Eclipse) para realizar testes de unidade, depuração e rastreamento
 - Usa o **sistema de arquivos local** em vez de **HDFS**
 - **Pseudo-Distribuído**
 - Todos os **daemons** do Hadoop são executados em JVMs separados em um único host
 - Simula como um cluster completo funcionaria
 - Útil para teste para simular a interação entre componentes em um cluster real usando uma única máquina
 - **Cluster Totalmente Distribuído (*Fully Distributed Cluster*)**
 - Executa daemons mestre e escravo em máquinas distintas
 - Modo de implantação típico para **sistemas de produção**

Instalação do Hadoop

- Hadoop foi **originalmente desenvolvido para Linux**
 - Já existe uma distribuição compatível com o Windows, mas pode apresentar problemas de compatibilidade com outros projetos de ecossistema Hadoop

Requisitos (clusters Hadoop de produção de média a grande escala)

- **Nós Mestre**

- 16 ou mais núcleos de CPU (de preferência 32)
- 128 GB ou mais de RAM (de preferência 256 GB)
- Configuração de disco rígido RAID (de preferência com unidades hot-swappable) - para tolerância a falhas
- Fontes de alimentação redundantes - para tolerância a falhas
- Bonded Gigabit Ethernet ou 10 Gigabit Ethernet

Requisitos (clusters Hadoop de produção de média a grande escala)

- **Nós escravos**
 - 16-32 núcleos de CPU
 - 64-512 GB de RAM
 - 12-24 discos rígidos de 1-4 TB em uma configuração JBOD
 - **JBOD (Just a Bunch of Disks)** - armazenamento conectado diretamente, **não em configuração RAID**; cada disco opera independentemente
 - **RAID não é recomendado para armazenamento em bloco em nós escravos!**
 - A velocidade de acesso é limitada pelo disco mais lento do array
 - JBOD é 30%-50% mais rápido do que RAID 0 no Hadoop
 - **Os nós escravos suportam falhas, portanto, eles não exigem o mesmo grau de tolerância a falhas que os nós mestres exigem**
 - **A capacidade de armazenamento em nós escravos não deve ser totalmente alocada!**
 - A falha de um nó pode impactar a rede devido à replicação de blocos

Alguns exemplos de configuração

- De <https://cwiki.apache.org/confluence/display/HADOOP2/PoweredBy>
- **Alibaba**
 - Um cluster de 15 nós dedicado a processar tipos de dados de negócios despejados do banco de dados e juntá-los. Esses dados são inseridos no iSearch (mecanismo de busca)
 - Cada nó possui 8 núcleos, 16G RAM e armazenamento de 1.4T
- **EBay**
 - 532 nós de cluster (8 * 532 núcleos, 5.3 PB) .
 - Uso intenso de Java MapReduce, Apache Pig, Apache Hive, Apache HBase
 - Usado para otimização de busca e pesquisa
- **Facebook**
 - Uso do Apache Hadoop para armazenar cópias de log interno e dimensionar fontes de dados e usá-lo como uma fonte para relatórios / análises e aprendizado de máquina.
 - Atualmente, possui 2 grandes clusters:
 - Um cluster de 1100 máquinas com 8800 núcleos e cerca de 12 PB de armazenamento bruto.
 - Um cluster de 300 máquinas com 2400 núcleos e cerca de 3 PB de armazenamento bruto.
 - Cada nó (commodity) possui 8 núcleos e 12 TB de armazenamento.

Alguns exemplos de configuração (parte 2)

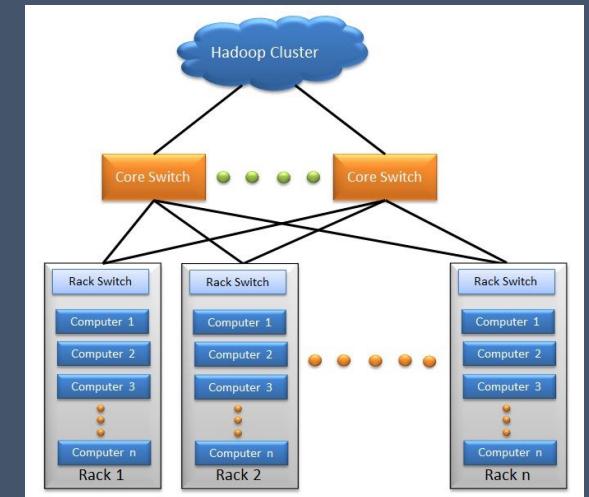
- De <https://wiki.apache.org/hadoop/PoweredBy>
- **Last FM**
 - 100 nós
 - Dual quad-core Xeon L5520 @ 2,27 GHz e L5630 @ 2,13 GHz, 24 GB de RAM, 8 TB (4x2 TB) / armazenamento de nó.
 - Usado para cálculo de gráficos, relatórios de royalties, análise de registro, teste A/B, agrupamento de datasets
 - Também usado para análise de features de áudio em grande escala com milhões de faixas
- **LinkedIn**
 - HP SL 170x com base em Westmere ~ 800, com núcleos 2x4, 24 GB de RAM, 6x2 TB SATA
 - ~ 1900 SuperMicro X8DTT-H baseado em Westmere, com 2x6 núcleos, 24 GB de RAM, 6x2 TB SATA
 - ~ 1400 SuperMicro baseado em Sandy Bridge com 2x6 núcleos, 32 GB de RAM, 6x2 TB SATA
 - “Usamos essas coisas para descobrir Pessoas que você pode conhecer e outros fatos divertidos.”
- **Spotify**
 - Uso do Apache Hadoop para geração de conteúdo, agregação de dados, relatórios, análises e até mesmo para gerar recomendações de música
 - Cluster de 1650 nós: 43.000 núcleos virtualizados, ~ 70 TB de RAM, ~ 65 PB de armazenamento
 - +20.000 jobs diários no Hadoop

Alguns exemplos de configuração (parte 3)

- De <https://wiki.apache.org/hadoop/PoweredBy>
- **Yahoo!**
 - Mais de 100.000 CPUs em > 40.000 computadores executando Hadoop
 - Maior cluster: 4.500 nós (2 * 4 CPU com disco de 4 * 1 TB e 16 GB de RAM)
 - Usado para apoiar a pesquisa de sistemas de anúncios e pesquisa na web
 - Também é usado para fazer testes de dimensionamento para oferecer suporte ao desenvolvimento do Apache Hadoop em clusters maiores

Considerações sobre a rede

- Implementações totalmente distribuídas consomem uma quantidade significativa de recursos de rede
 - Controle de mensagens, atualizações de status e heartbeats, relatórios de bloco, embaralhamento (shuffling) de dados e replicação de bloco
- Implantação local
 - Sub-rede privada com switches dedicados
 - Switches redundantes de núcleo e TOR (topo do rack)
 - Utilização de STP (Spanning Tree Protocol) para evitar loops
- **Resolução de nome de host entre nós**
 - Configuração de DNS (padrão e reverso) ou arquivo de hosts
- **Sincronização de tempo (NTP - Network Time Protocol)**
 - Usado por alguns componentes (por exemplo, Kerberos)



Requisitos de software

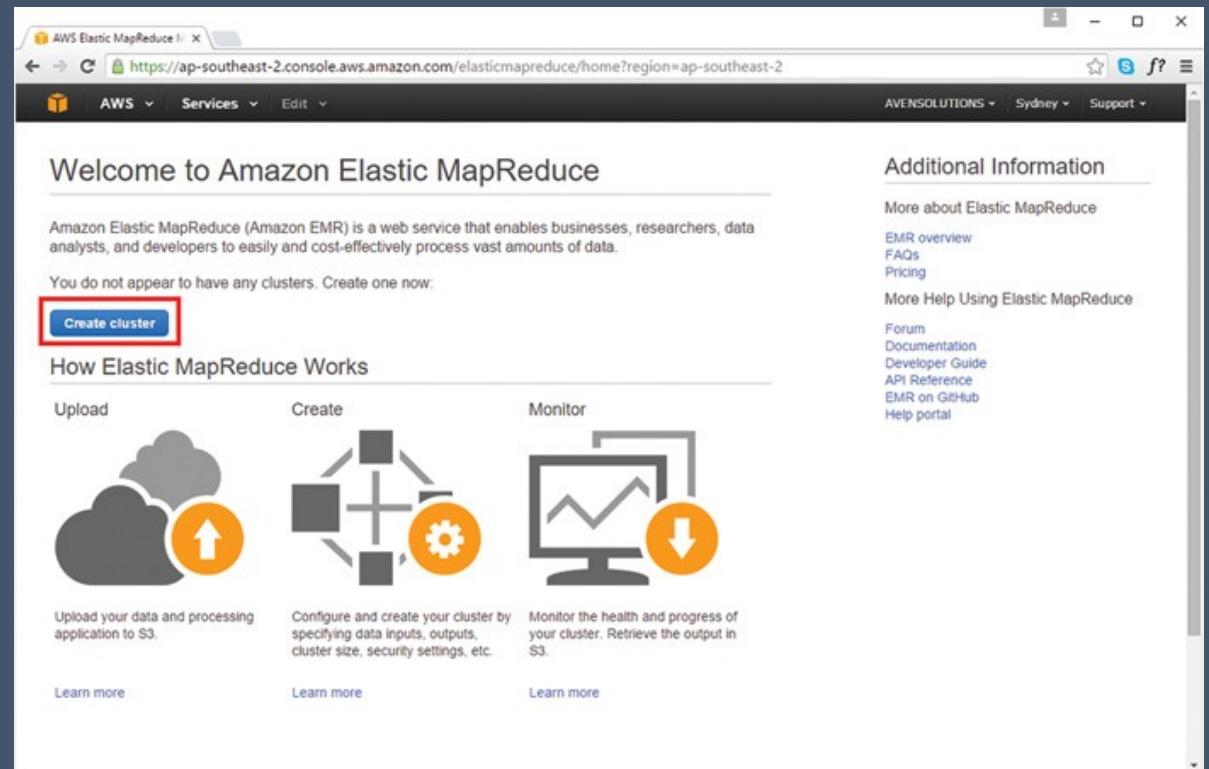
- Hadoop é quase totalmente escrito em Java
- Requisitos
 - Java 8 (versões 2.7.x a 2.10.xe versões 3.0.x a 3.2.x)
 - A versão 3.3 e superior também oferecem suporte a Java 11 (somente para runtime - a compilação deve usar Java 8)
- Versões OpenJDK e Oracle suportadas
- Algumas versões podem apresentar problemas!
- Site com JDKs testados
 - <https://cwiki.apache.org/confluence/display/HADOOP/Hadoop+Java+Versions>
- Outros projetos de ecossistema podem ter outros requisitos específicos
 - Por exemplo, o Apache Spark requer Scala e Python

Instalando o Hadoop On-premises

- Hadoop não comercial
 - Instalação a partir dos builds da Apache - <http://hadoop.apache.org>
- Distribuições Comerciais
 - Cloudera (<http://www.cloudera.com/>)
 - **Cloudera CDH** - <https://www.cloudera.com/downloads/cdh.html>
 - Requer assinatura para versões atuais
 - **Cloudera Manager** (software proprietário) - fornece um console de gerenciamento e estrutura para implantar, gerenciar e monitorar clusters Hadoop
 - **Hortonworks Sandbox** - VM com modo pseudo-distribuído Hadoop - <https://www.cloudera.com/downloads/hortonworks-sandbox.html>
 - Hortonworks Data Platform (distribuição com pilha Hadoop completa e componentes de ecossistema selecionados)
 - Usa **Apache Ambari** para implantação e gerenciamento
 - MapR (<https://www.mapr.com/>) - Adquirido pela HPE
 - **MapR Sandbox** - VM com modo pseudo-distribuído Hadoop - <https://www.mapr.com/products/mapr-sandbox-hadoop/download>
 - **MapR Control System (MCS)** - console central para configurar, monitorar e gerenciar clusters MapR

Hadoop na nuvem

- Amazon Web Services
 - EMR (Elastic MapReduce) - plataforma Hadoop-as-a-Service da Amazon
 - Permite a criação de clusters com um número específico de nós, tipos de instância de nó (EC2), distribuição Hadoop e aplicativos de ecossistema adicionais
 - Usa S3 para ler / escrever dados
 - Provisionado sob demanda
 - <https://aws.amazon.com/emr/>



Hadoop na nuvem

- Microsoft Azure HDInsight
 - <https://azure.microsoft.com/en-us/services/hdinsight/>

The image shows two screenshots of the Microsoft Azure portal.

Left Screenshot (Create View):

- 1. A red box highlights the "NEW" button in the top-left corner of the "Create" view.
- 2. A red box highlights the "Data + Analytics" category in the "Browse" section.
- 3. A red box highlights the "HDInsight" service card, which describes it as "Microsoft's cloud-based Big Data service. Apache Hadoop and other popular Big Data solutions."

Right Screenshot (Cluster Overview):

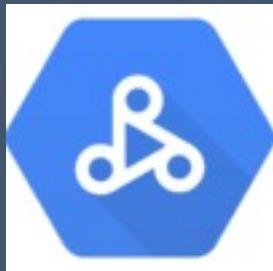
A cluster named "myhdinsightcluster" is shown in the "RUNNING" state. The cluster details are as follows:

- Resource group: myresourcegroupsouth
- Status: Running
- Location: South Central US
- Subscription name: [REDACTED]
- Subscription id: [REDACTED]
- URL: myhdinsightcluster.azurehdinsight.net
- Cluster Type: hadoop
- Operating system: Linux
- Worker Nodes: Standard_D12, Small
- Head Node: Standard_D12

At the bottom right of the cluster details, there is a link labeled "All settings →".

Hadoop na nuvem

- Google Cloud Dataproc
 - <https://cloud.google.com/dataproc/>



Cloud Dataproc

Create a cluster

Name: example-cluster

Region: global Zone: us-central1-a

Master node
Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type: n1-standard-4 (4 vCPU, 15.0 GB ...) Cluster mode: Standard (1 master, N workers)

Primary disk size (minimum 10 GB): 500 GB

Worker nodes
Each contains a YARN NodeManager and a HDFS DataNode. The HDFS replication factor is 2.

Machine type: n1-standard-4 (4 vCPU, 15.0 GB ...) Nodes (minimum 2): 2

Primary disk size (minimum 10 GB): 500 GB Local SSDs (0-8): 0 x 375 GB

YARN cores: 8 YARN memory: 24.0 GB

Preemptible workers, bucket, network, version, initialization, & access options

Create Cancel

Equivalent REST or command line

Ciências de Dados e Inteligência Artificial

Gerência de Infraestrutura para
Big Data

HADOOP

Prof. Tiago Ferreto

tiago.ferreto@pucrs.br

