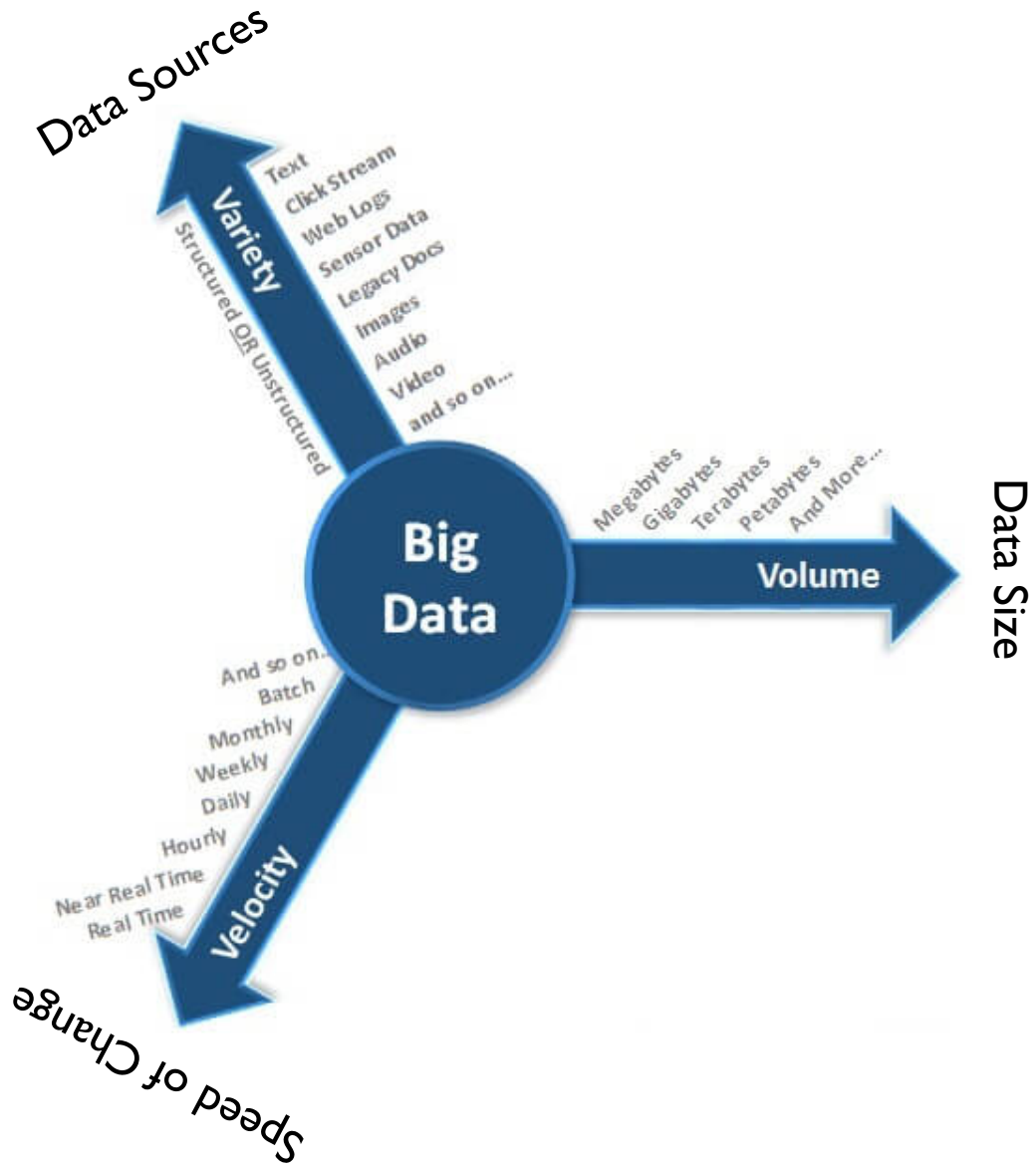
BIG DATA

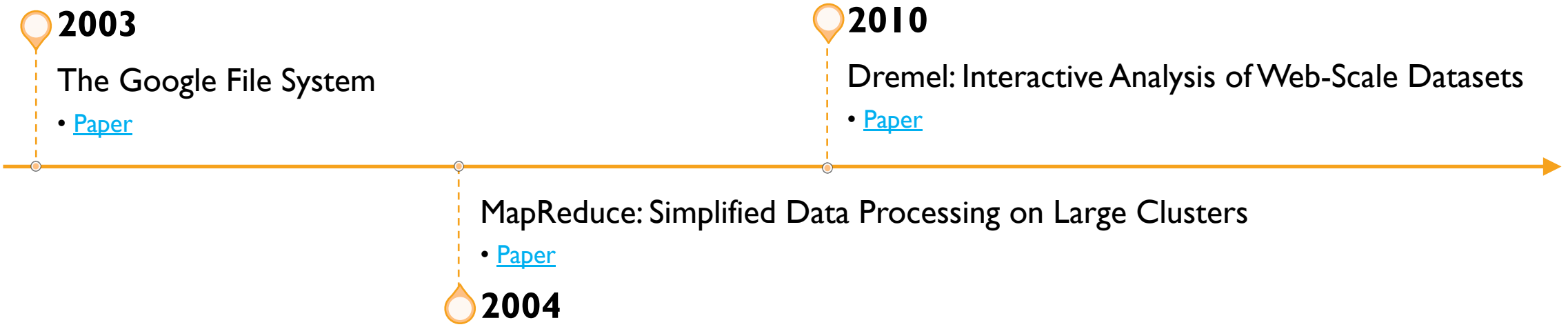
CRISTOFER WEBER

- Cientista de Dados na Plataforma Digital do Sicredi
- Professor na Especialização em Data Science e Big Data da UniRitter.
- Mestre em Ciência da Computação pela PUC-RS na área de Inteligência Artificial
- Membro organizador dos Meetups "UserRS - RS R User Group" e "RS Data Science Meetup".
- Coordenador das trilhas de Data Science e Big Data no TDC (POA & Floripa)
- 20 anos de experiência com tecnologia, em papéis ligados ao desenvolvimento de software, estratégia e ciência de dados.
- Big Data & Data Science desde 2012

INTRODUÇÃO


Data Sources
Variety
Structured OR Unstructured
Text
Click Stream
Web Logs
Sensor Data
Legacy Docs
Images
Audio
Video
and so on...
Big Data
Volume
Megabytes
Gigabytes
Terabytes
Petabytes
And More...
Data Size
Velocity
And so on...
Batch
Monthly
Weekly
Daily
Hourly
Near Real Time
Real Time
Speed of Change

ORIGEM


2003
The Google File System
• Paper
2004
MapReduce: Simplified Data Processing on Large Clusters
• Paper
2010
Dremel: Interactive Analysis of Web-Scale Datasets
• Paper

GOOGLE

GOOGLE

- A scalable distributed file system for large distributed data-intensive applications

  #scalable #distributed #large #data-intensive

- … shares many of the same goals as previous distributed file systems such as performance, scalability, reliability, and availability

  #performance #reliability #availability

- Its design has been driven by observations of our application workloads and technological environment that reflect a marked departure from some earlier file system design assumptions.

  #workloads #technological-environment

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

1. Component failures are the norm rather than the exception
2. Files are huge by traditional standards. Multi-GB files are common. Each file typically contains many application objects such as web documents
3. Most files are mutated by appending new data rather than overwriting existing data
4. Co-designing the applications and the file system API benefits the overall system by increasing our flexibility.

GOOGLE

- Programming Model
- Implementation
- Runtime System
- Partitioning
- Scheduling Execution
- Large cluster of commodity machines
- Handling machine failures
- Managing inter-machine communication

- Map, Combiner, Partitioner, Shuffle, Sort, Reduce

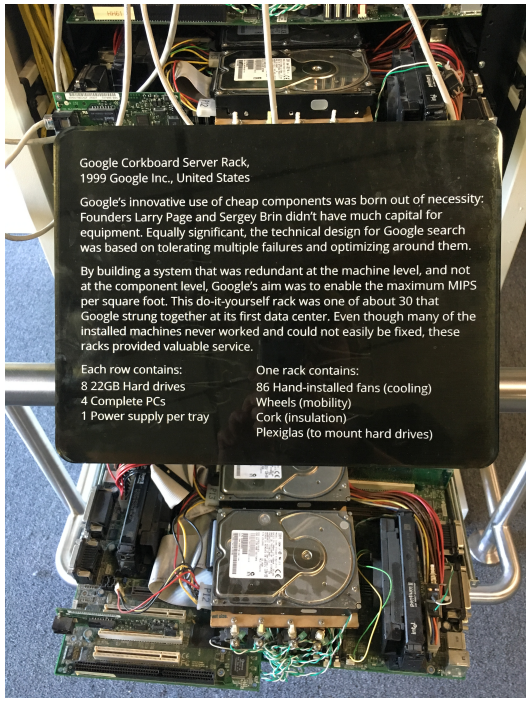
- MapReduce is a programming model and an associated implementation for processing and generating large data sets. **Users** specify:
  - A map function: processes a key/value pair to generate a set of intermediate key/value pairs,
  - A reduce function: merges all intermediate values associated with the same intermediate key.
- Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines.
- The runtime system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication.

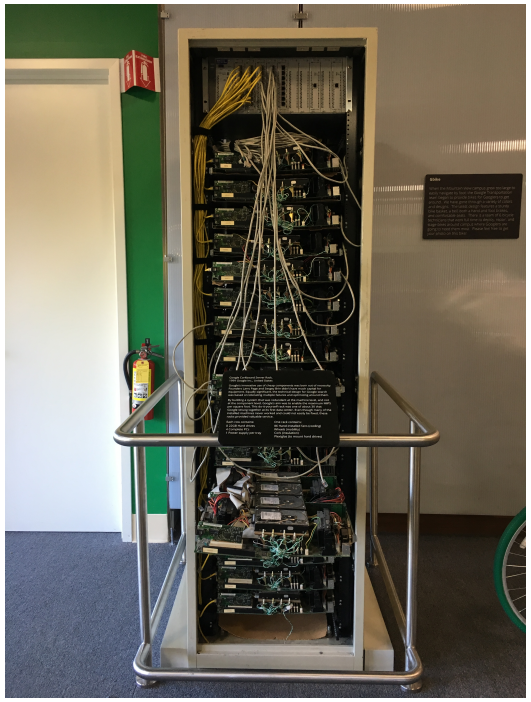
MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.


Google Corkboard Server Rack,
1999 Google Inc., United States
Google's innovative use of cheap components was born out of necessity: Founders Larry Page and Sergey Brin didn't have much capital for equipment. Equally significant, the technical design for Google search was based on tolerating multiple failures and optimizing around them.
By building a system that was redundant at the machine level, and not at the component level, Google's aim was to enable the maximum MIPS per square foot. This do-it-yourself rack was one of about 30 that Google strung together at its first data center. Even though many of the installed machines never worked and could not easily be fixed, these racks provided valuable service.
Each row contains:
8 22GB Hard drives
4 Complete PCs
1 Power supply per tray
One rack contains:
86 Hand-installed fans (cooling)
Wheels (mobility)
Cork (insulation)
Plexiglas (to mount hard drives)



CLUSTER

GOOGLE

- Dremel is a **scalable**, **interactive ad-hoc query** system for **analysis** of **read-only** nested data.
- By combining multi-level execution trees and **columnar data layout**, it is capable of running aggregation queries over trillion-row tables in seconds.
- The system **scales** to **thousands of CPUs** and **petabytes of data.**
- **Dremel has been in production since 2006** and has thousands of users within Google
- Dremel builds on ideas from web search and parallel DBMSs:
  - Its architecture borrows the concept of a serving tree used in distributed search engine
  - Dremel **provides a high-level, SQL-like language** to express ad hoc queries. In contrast to layers such as Pig and Hive, **it executes queries natively without translating them into MR jobs**.

APACHE HADOOP

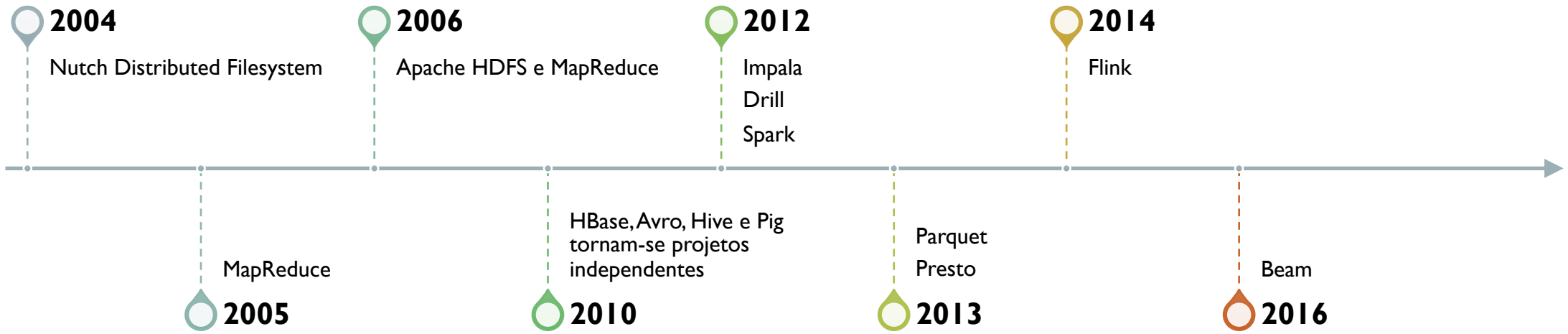
- **Hadoop** é uma plataforma de computação distribuída voltada para processamento de grandes volumes de dados, com atenção a tolerância a falhas.
- Foi inspirada no MapReduce e no GoogleFS

- Composto pelos seguintes módulos:
  - **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
  - **Hadoop YARN**: A framework for job scheduling and cluster resource management.
  - **Hadoop Common**: The common utilities that support the other Hadoop modules.
  - **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

APACHE HADOOP

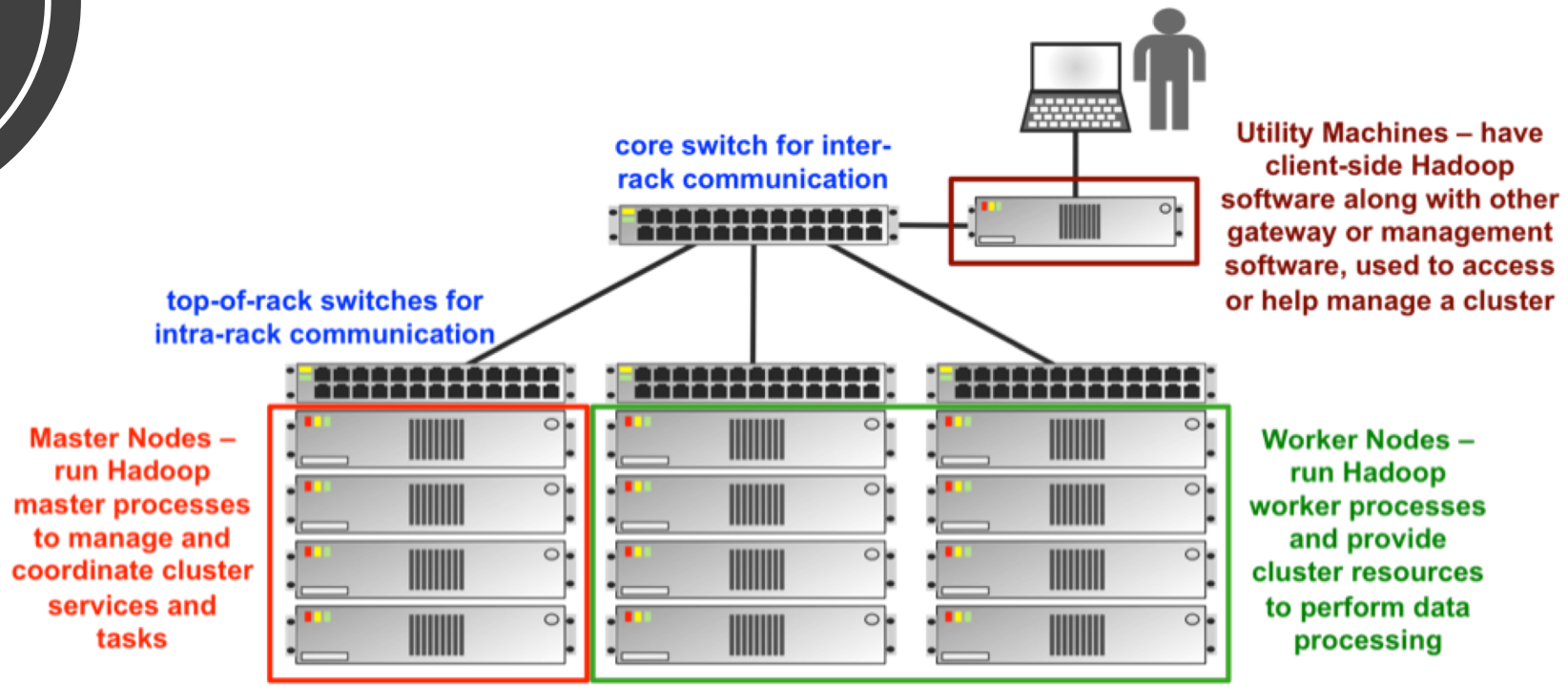
Projetos relacionados

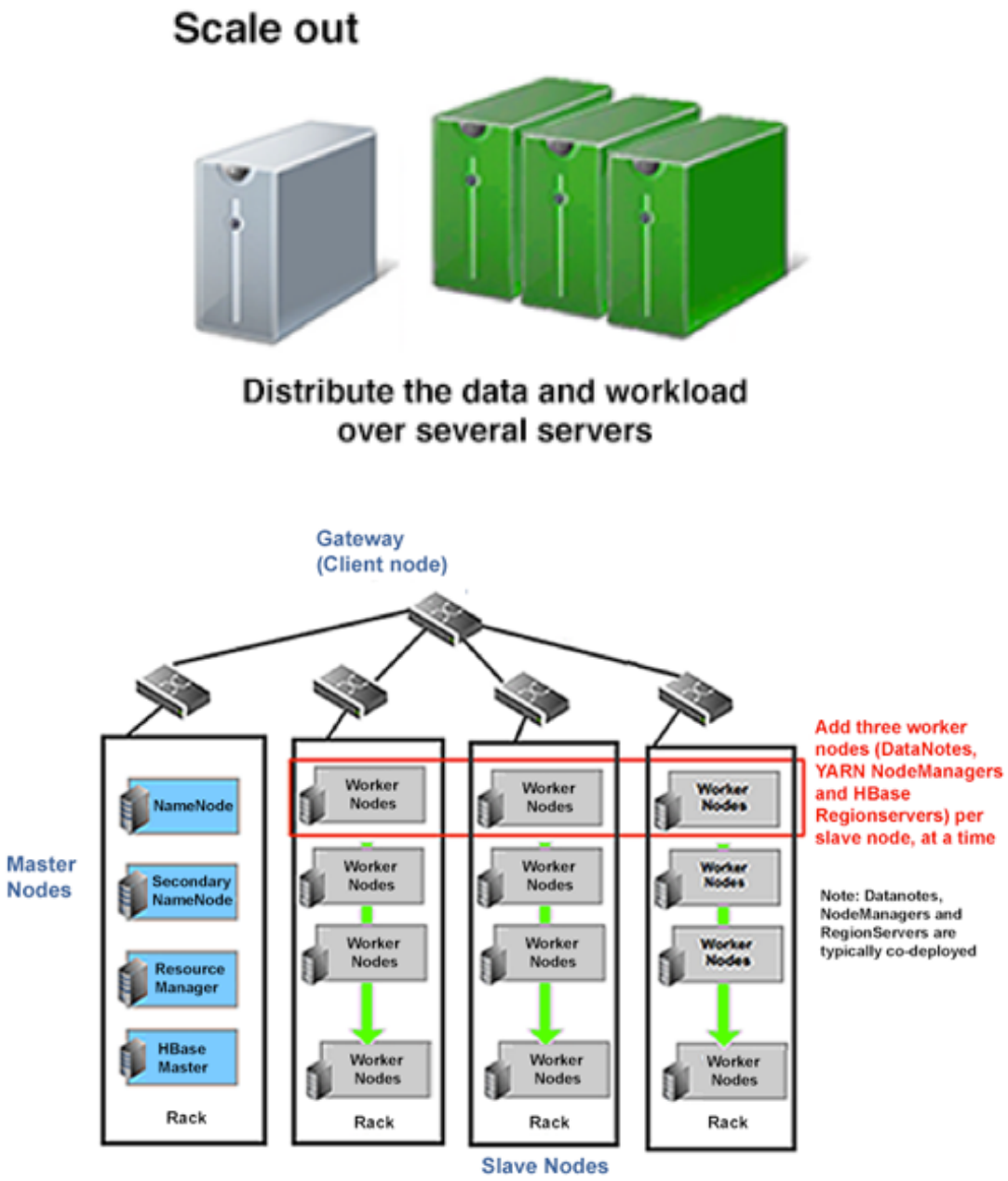
- **Avro**: A data serialization system.
- **HBase**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **ZooKeeper**: A high-performance coordination service for distributed applications.



OPEN SOURCE (HADOOP & OUTROS)

HADOOP CLUSTER

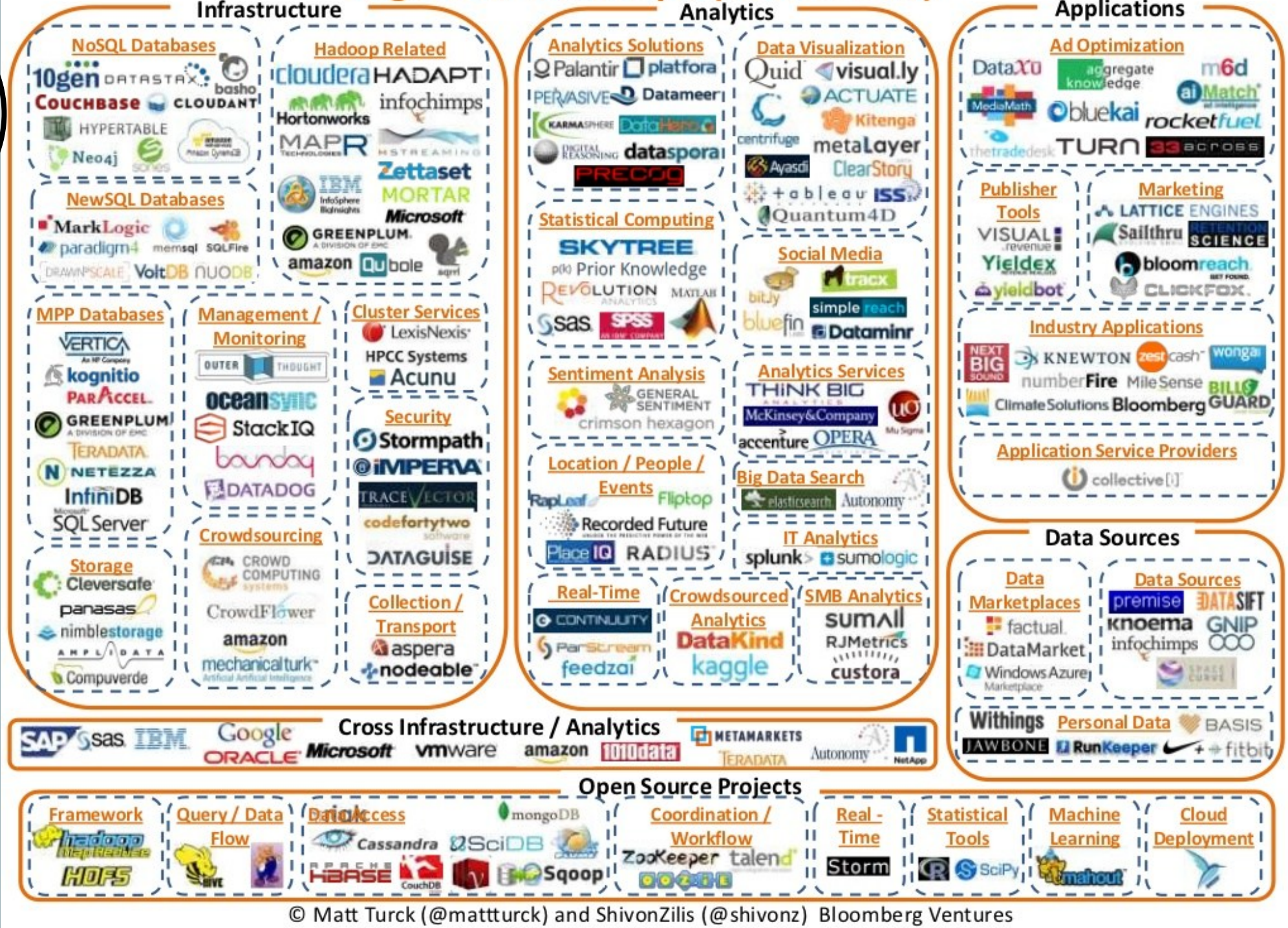




HADOOP CLUSTER

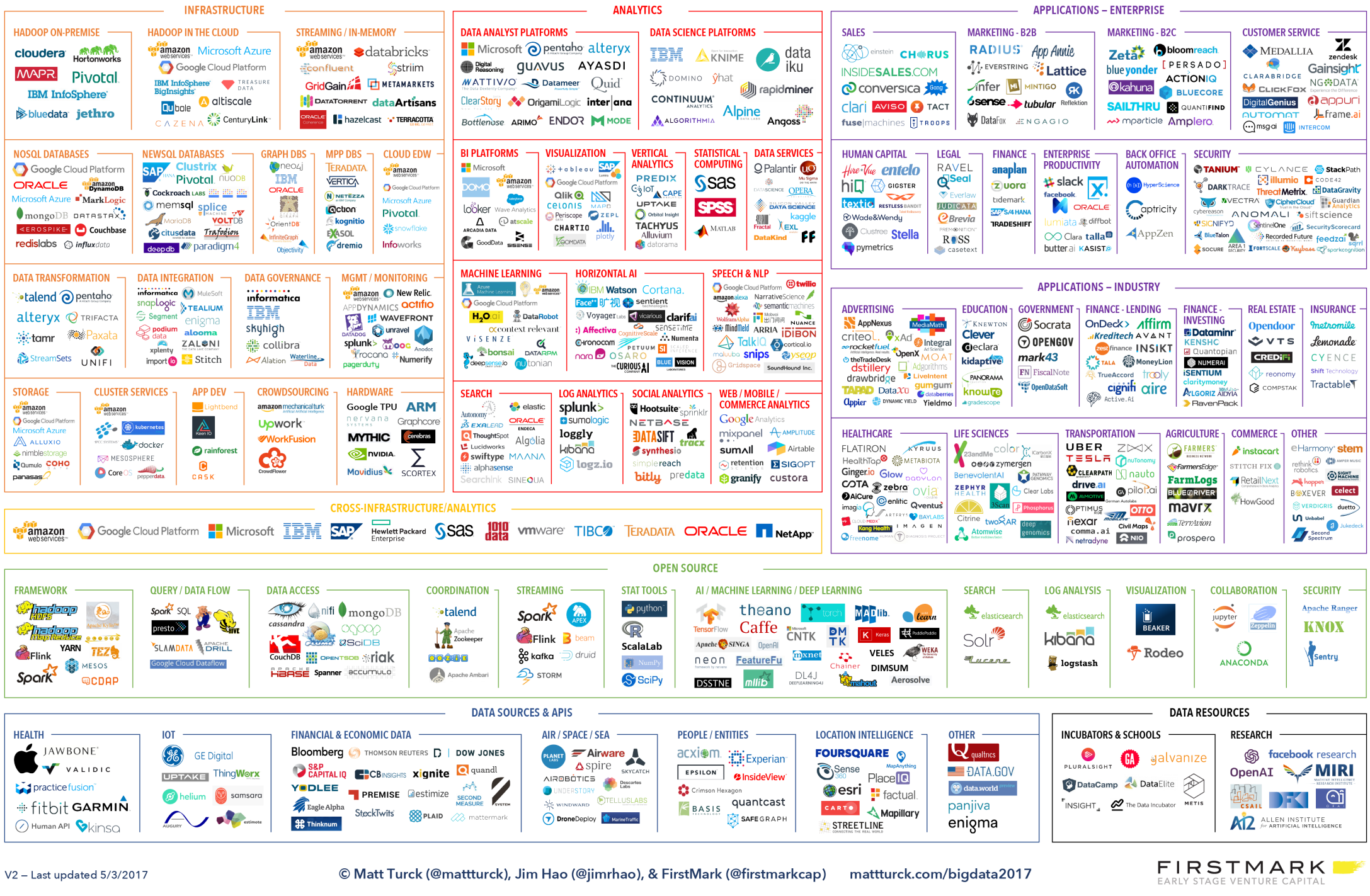
- **Gateway Node**
  - Ponto de comunicação externa com o cluster
- **Master Nodes**
  - Serviços de gerenciamento e coordenação
- **Worker Nodes**
  - Serviços de processamento

20
12



© Matt Turck (@mattturck) and ShivonZilis (@shivonz) Bloomberg Ventures

2017



© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark (@firstmarkcap)

mattturck.com/bigdata2017

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

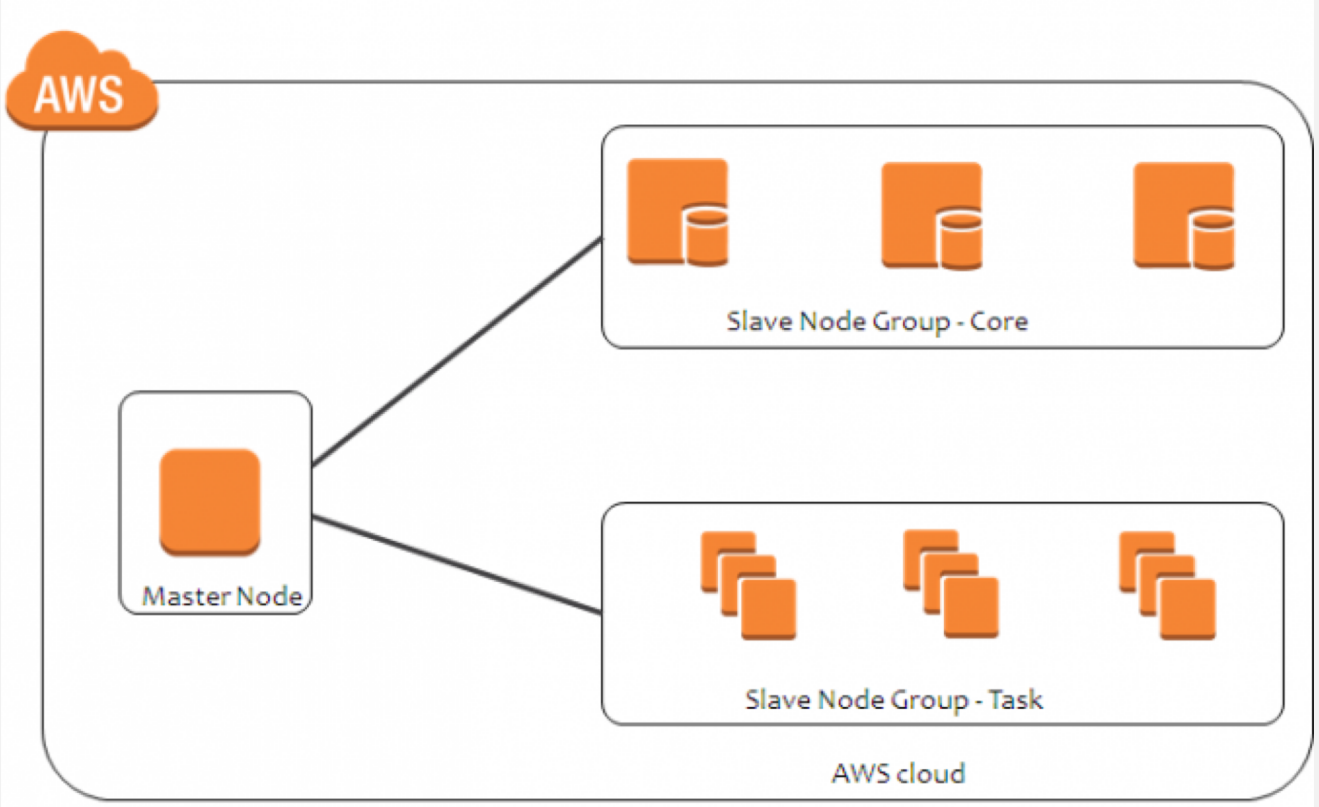
PÓS-HADOOP

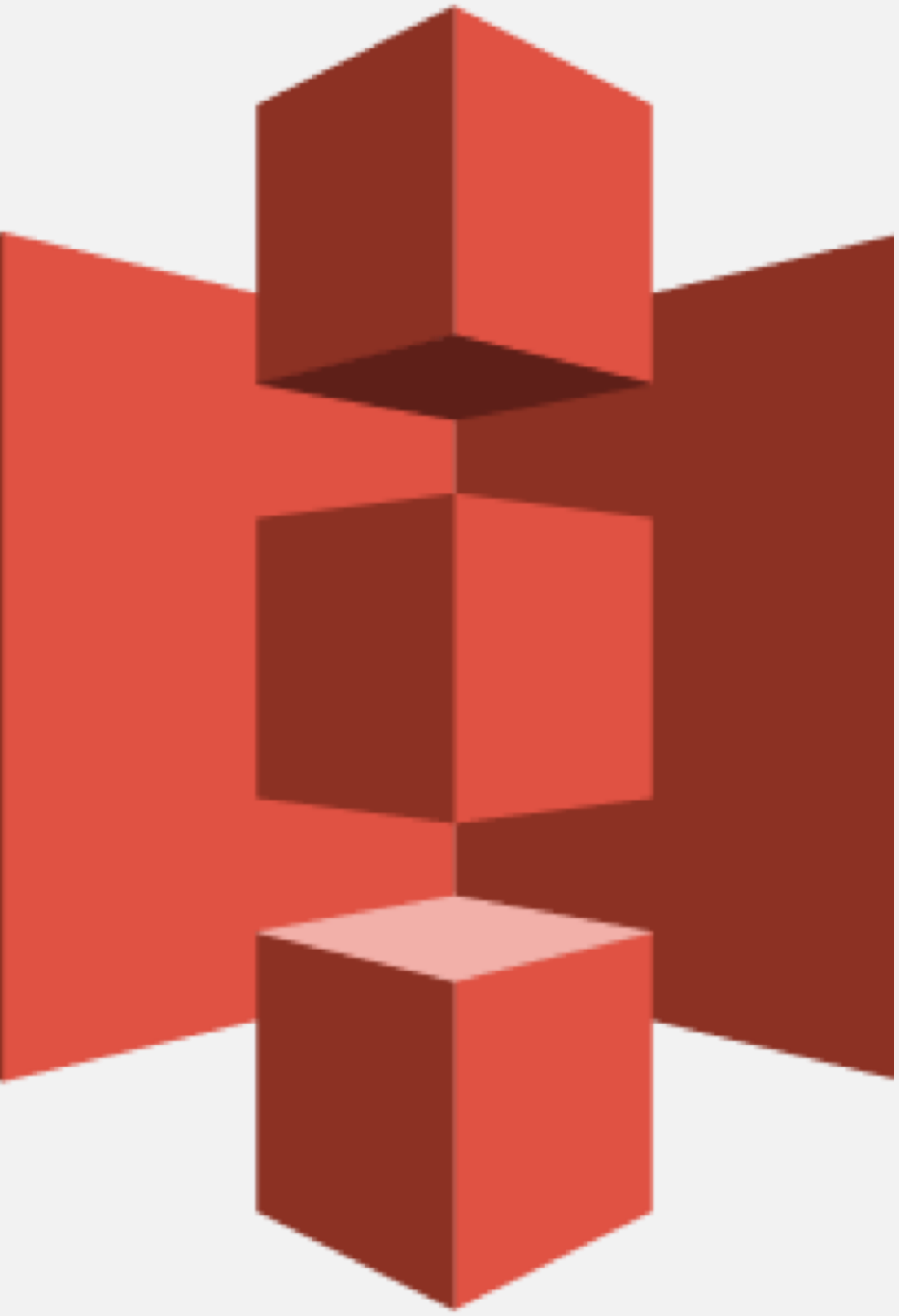
CLOUD COMPUTING

- **IaaS** (Infrastructure as a Service)
  - O hardware e os serviços básicos (Sistema Operacional, Rede) são disponibilizados pelo provedor.
  - A instalação e configuração de todos os demais serviços é de responsabilidade do contratante
  - Exemplo: Amazon EC2
- **PaaS** (Platform as a Service)
  - O hardware, os serviços básicos, e os serviços de plataforma são disponibilizados e pré-configurados pelo provedor.
  - O contratante é responsável pela escolha da capacidade de hardware e dos serviços de plataforma que estarão disponíveis em cada servidor
  - Exemplo: Amazon Elastic MapReduce

EMR CLUSTER

- **Master node**: A node that manages the cluster by running software components to coordinate the distribution of data and tasks among other nodes.
  - The master node tracks the status of tasks and monitors the health of the cluster.
- **Core node**: A slave node with software components that run tasks and store data in the Hadoop Distributed File System (HDFS) on your cluster.
- **Task node**: A slave node with software components that only run tasks. Task nodes are optional.



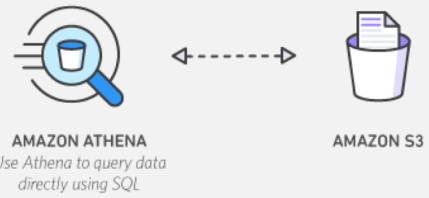


CLOUD COMPUTING

- **Storage (serverless)**
  - Object Stores como o Amazon S3 evoluíram de simples APIs de armazenamento e recuperação de dados, viabilizando a análise dos dados sem a necessidade de movimentá-los para um sistema de análises separado.
  - Em uma arquitetura Big Data nativa na nuvem, os benefícios do HDFS são mínimos e não justificam a complexidade computacional.

CLOUD COMPUTING

- **Serverless** (Servidores de outra pessoa ☺)
- Serviços de propósito específico que não requerem a contratação de infraestrutura para seu processamento.
- O provedor gerencia toda a infra subjacente e o contratante paga pelo uso.
- Exemplos:
  - Amazon Athena para processamento de consultas interativas



  - Amazon SageMaker para treino e publicação de modelos de ML

Amazon Athena

Amazon SageMaker

PÓS-HADOOP

- **Apache Spark**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including **ETL**, **machine learning**, stream processing, and graph computation.
- **Apache Tez**: A generalized data-flow programming framework which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases, being adopted by Hive, Pig to replace Hadoop MapReduce as the underlying execution engine.
- **Apache Flink:** an open-source stream processing framework for **distributed, high-performing, always-available,** and **accurate** data streaming applications.

CAPACIDADE COMPUTACIONAL

- Redes foram de 1GB para 10GB
- Há disponibilidade de servidores EC2 com 2TB de memória
- Intel já fornece processadores com 28 Cores / 56 Threads
- Novas arquiteturas especializadas:
  - GPUs, FPGAs, Nervana NNP
- Storage SSD de baixa latência

NOVAS TECNOLOGIAS

- Bancos de dados analíticos baseados em GPU
  - brytlyt
  - MapD*
  - Kinetica
- Aceleração de processamento de Bancos de dados com FPGA
  - Swarm64 Data Accelerator