Hadoop

FILIPPO VALIANTE FILHO

O que é Hadoop

- Um projeto Open Source da Fundação Apache para computação distribuída, escalável e confiável: http://hadoop.apache.org
  - The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. [Apache]
- Ecossistema ou sistema operacional para Big Data.
- Framework de software para armazenagem, processamento e análise de "big data". [Cloudera]
- Apache Hadoop® é uma estrutura em código aberto para armazenamento e processamento distribuídos de grandes conjuntos de dados em hardware simples. O Hadoop permite que as empresas obtenham insights de quantidades enormes de dados estruturados e desestruturados de forma rápida. [HortonWorks]

O que é Hadoop (2)

- Distribuído
- Escalável
- Flexível
- Tolerante a falhas
- Open source
- Baixo custo (open source e hardware commodity)

História

Happy 10th Birthday, Apache Hadoop

In 2016, Apache Hadoop and its community turns 10 years old. To help celebrate this momentous event throughout the year, we'll be spotlighting the community's accomplishments over the last decade in multiple ways —so bookmark this page and follow the #Hadoop10 hashtag for updates.

Also, join us at **Strata + Hadoop World 2016 in London** for an official celebration of Hadoop's birthday, and for extended conversation about what the Hadoop ecosystem does next.

http://www.cloudera.com/promos/hadoop10.html

A BRIEF HISTORY OF APACHE HADOOP

WHAT IS HADOOP?

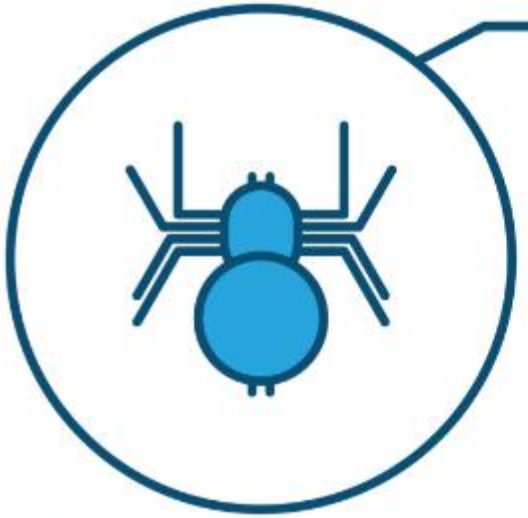
Hadoop — officially born on January 28, 2006 — is an ecosystem of open source components that fundamentally changed the way enterprises store, process, and analyze data. Unlike traditional systems, Hadoop enables multiple types of analytic workloads to run on the same data and at the same time.

Lines of code currently in core Hadoop:

1.7 MILLION+

Total commits to core Hadoop since 2006:

12,000+



OCTOBER 2002:

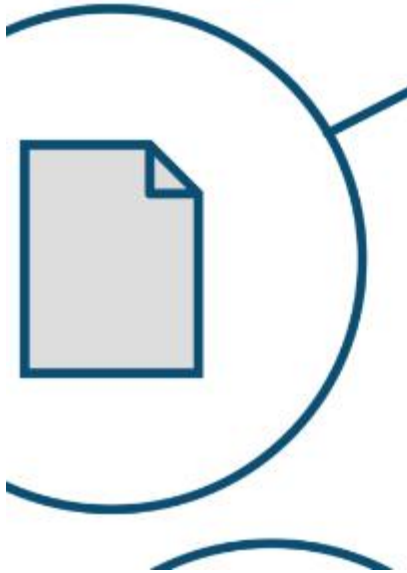
Doug Cutting and Mike Cafarella create Nutch, an open source web crawler

OCTOBER 2003:

Google publishes its “The Google File System” paper

JUNE 2004:

Cutting and Cafarella implement features in Nutch that will eventually become core Hadoop



OCTOBER 2004:

Google publishes the original MapReduce paper

FEBRUARY 2005:

Cafarella spearheads Nutch’s MapReduce implementation

JANUARY 2006:

Doug Cutting joins Yahoo!

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

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 and executed on a large cluster of commodity machines. The run-time 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. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of MapReduce programs have been implemented and upwards of one thousand MapReduce jobs are executed on Google's clusters every day.

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 computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution and load balancing in a library. Our abstraction is inspired by the *map* and *reduce* primitives present in Lisp and many other functional languages. We realized that most of our computations involved applying a *map* operation to each logical "record" in our input in order to compute a set of intermediate key/value pairs, and then applying a *reduce* operation to all the values that shared the same key, in order to combine the derived data appropriately. Our use of a functional model with user-specified map and reduce operations allows us to parallelize large computations easily and to use re-execution as the primary mechanism for fault tolerance.

The major contributions of this work are a simple and powerful interface that enables automatic parallelization

JANUARY 2006:

Cutting starts new subproject by carving Hadoop code from Nutch

- The name Hadoop comes from a plush-toy elephant owned by Cutting's young son

MARCH 2006:

Yahoo! creates its first Hadoop research cluster

APRIL 2006:

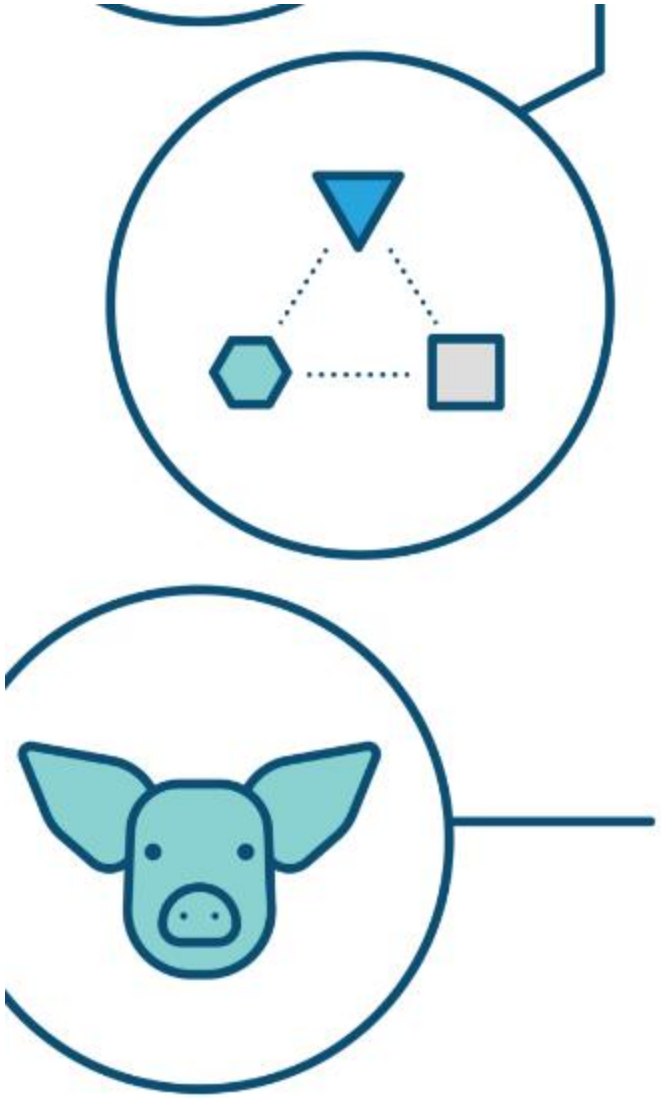
First Apache release of Hadoop

NOVEMBER 2006:

Google publishes the original Bigtable paper, which will eventually inspire the creation of Apache HBase

Hadoop was created by Doug Cutting and Mike Cafarella in 2005. It was originally developed to support distribution for the Nutch search engine project. Doug, who was working at Yahoo! at the time and is now Chief Architect of Cloudera, named the project after his son's toy elephant. Cutting's son was 2 years old at the time and just beginning to talk. He called his beloved stuffed yellow elephant "Hadoop" (with the stress on the first syllable). Now 12, Doug's son often exclaims, "Why don't you say my name, and why don't I get royalties? I deserve to be famous for this!"

http://opensource.com/life/14/8/intro-apache-hadoop-big-data



JANUARY 2008:

Hadoop becomes a top-level Apache project

FEBRUARY 2008:

Yahoo! announces launch of the world's then-largest Hadoop application

JUNE 2008:

Hive, Hadoop's first SQL access framework, becomes a new sub-project

AUGUST 2008:

Founding of Cloudera, the first company to commercialize Hadoop

SEPTEMBER 2008:

Initial Apache release of Pig, the first high-level non-SQL framework for Hadoop

JUNE 2009:

Publication of Tom White's Hadoop: The Definitive Guide

AUGUST 2009:

Doug Cutting joins Cloudera as its chief architect

OCTOBER 2009:

Inaugural Hadoop World convenes in New York City

MARCH 2012:

HDFS HA, a significant step forward for enterprise adoption, merges into the Hadoop trunk

AUGUST 2012:

YARN, another milestone for enterprise adoption, becomes a Hadoop sub-project

OCTOBER 2012:

Impala, the first native MPP analytic database for Hadoop, joins the ecosystem

FEBRUARY 2014:

Apache Spark, the emerging default execution engine for Hadoop, becomes a top-level Apache project

Em 2011, Rob Bearden firmou parceria com a Yahoo! para fundar a Hortonworks com 24 engenheiros da equipe original Hadoop. [HortonWorks]

Contributors to core Hadoop since 2006:

800+

Number of components in Hadoop ecosystem

Quem usa o Hadoop?

Lista extensa (e não muito atualizada?) em: http://wiki.apache.org/hadoop/PoweredBy

Yahoo!

More than 100,000 CPUs in >40,000 computers running Hadoop

*Our biggest cluster: 4500 nodes (2*4cpu boxes w 4*1TB disk & 16GB RAM)*

- *Used to support research for Ad Systems and Web Search*
- *Also used to do scaling tests to support development of Apache Hadoop on larger clusters*

>60% of Hadoop Jobs within Yahoo are Apache Pig jobs.

Quem usa o Hadoop? (2)

Twitter

We use Apache Hadoop to store and process tweets, log files, and many other types of data generated across Twitter. We store all data as compressed LZO files.

We use both Scala and Java to access Hadoop's MapReduce APIs

We use Apache Pig heavily for both scheduled and ad-hoc jobs, due to its ability to accomplish a lot with few statements.

We employ committers on Apache Pig, Apache Avro, Apache Hive, and Apache Cassandra, and contribute much of our internal Hadoop work to opensource (see hadoop-lzo)

For more on our use of Apache Hadoop, see the following presentations: Hadoop and Pig at Twitter and Protocol Buffers and Hadoop at Twitter

Quem usa o Hadoop? (3)

LinkedIn

We have multiple grids divided up based upon purpose.

Hardware:

- ~800 Westmere-based HP SL 170x, with 2x4 cores, 24GB RAM, 6x2TB SATA
- ~1900 Westmere-based SuperMicro X8DTT-H, with 2x6 cores, 24GB RAM, 6x2TB SATA
- ~1400 Sandy Bridge-based SuperMicro with 2x6 cores, 32GB RAM, 6x2TB SATA

Software:

- RHEL 6.3
- Sun JDK 1.6.0_32
- Apache Hadoop 0.20.2+patches and Apache Hadoop 1.0.4+patches
- Pig 0.10 + DataFu
- Azkaban and Azkaban 2 for scheduling
- Apache Hive, Apache Avro, Apache Kafka, and other bits and pieces...

We use these things for discovering People You May Know and other fun facts.

Quem usa o Hadoop? (4)

Facebook

We use Apache Hadoop to store copies of internal log and dimension data sources and use it as a source for reporting/analytics and machine learning.

Currently we have 2 major clusters:

- *A 1100-machine cluster with 8800 cores and about 12 PB raw storage.*
- *A 300-machine cluster with 2400 cores and about 3 PB raw storage.*
- *Each (commodity) node has 8 cores and 12 TB of storage.*

- *We are heavy users of both streaming as well as the Java APIs. We have built a higher level data warehousing framework using these features called Hive (see the http://hadoop.apache.org/hive/). We have also developed a FUSE implementation over HDFS.*

Para que serve?

Tipo de análises comuns com Hadoop:

- Text mining
- Index building
- Graph creation and analysis
- Pattern recognition
- Collaborative filtering
- Prediction models
- Sentiment analysis
- Risk assessment

Características comuns dos problemas a serem processados no Hadoop?

Natureza dos dados (os "Vs" do Big Data)

- Volume ("petabyte scale")
- Velocity
- Variety

Natureza das análises

- Batch processing
- Parallel execution
- Distributed data

[Cloudera]

Distribuições Hadoop

Única 100%
Open Source

cloudera

Pivotal

MAPR

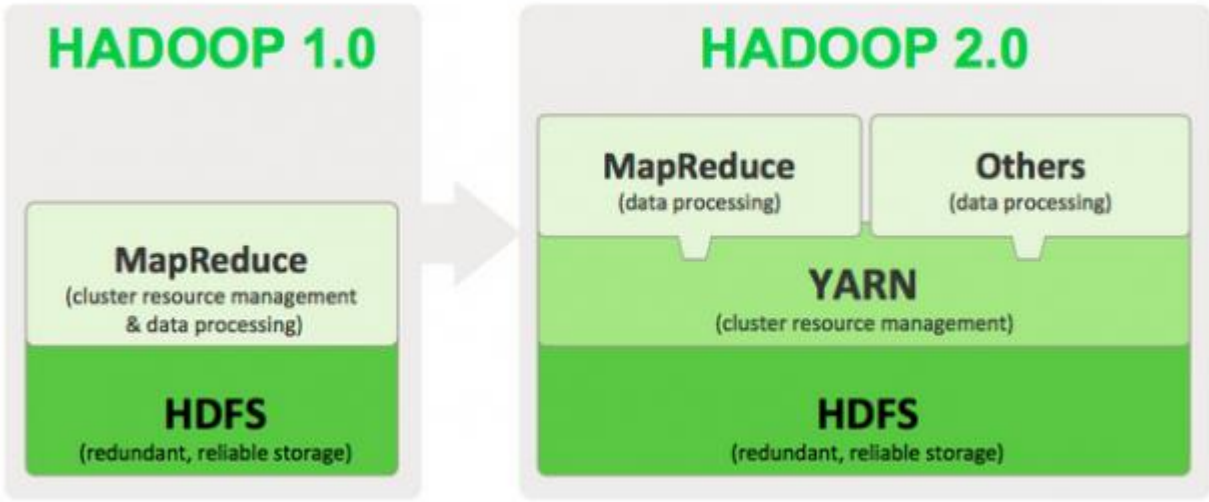
IBM Open Platform

Estrutura (1)

Core Components:

- HDFS (Hadoop Distributed File System)
  - Armazena dados no cluster
- MapReduce
  - Processa dados no cluster

A primeira versão do Hadoop se resumia a esses 2 componentes!



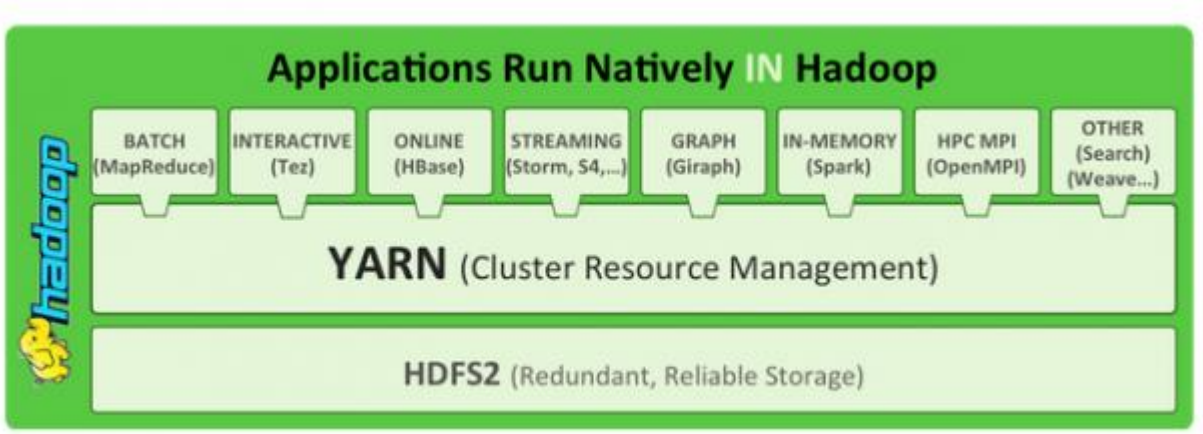
Estrutura (2)

A versão 2 do Hadoop tem os seguintes componentes:

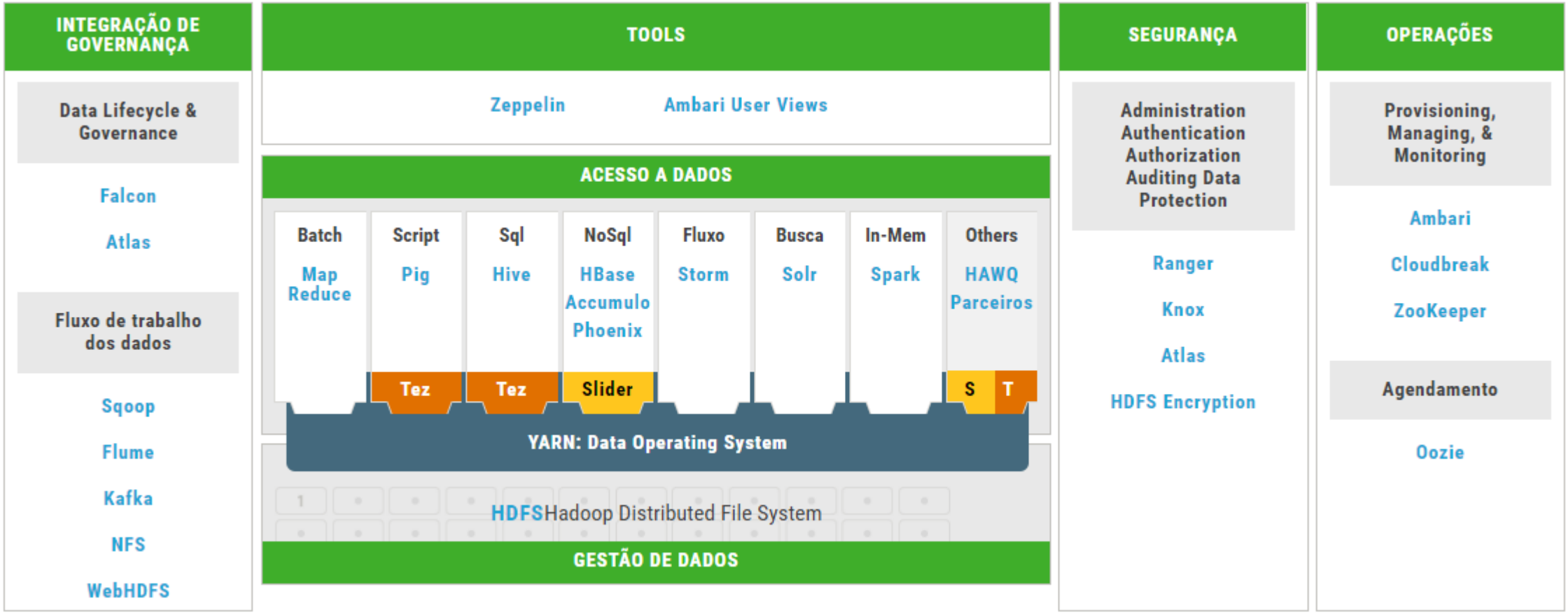
- **Hadoop Common**
  - The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**
  - A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**
  - A framework for job scheduling and cluster resource management.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

Estrutura (2)

Hadoop 2 suporta diversas outras aplicações além do MapReduce:



Estrutura (3)


INTEGRAÇÃO DE GOVERNANÇA
Data Lifecycle & Governance
Falcon
Atlas
Fluxo de trabalho dos dados
Sqoop
Flume
Kafka
NFS
WebHDFS
TOOLS
Zeppelin
Ambari User Views
ACESSO A DADOS
Batch
Map Reduce
Script
Pig
Sql
Hive
NoSql
HBase
Accumulo
Phoenix
Fluxo
Storm
Busca
Solr
In-Mem
Spark
Others
HAWQ
Parceiros
Tez
Tez
Slider
S
T
YARN: Data Operating System
HDFSHadoop Distributed File System
GESTÃO DE DADOS
SEGURANÇA
Administration
Authentication
Authorization
Auditing Data Protection
Ranger
Knox
Atlas
HDFS Encryption
OPERAÇÕES
Provisioning, Managing, & Monitoring
Ambari
Cloudbreak
ZooKeeper
Agendamento
Oozie

Estrutura (4)


GERENCIAMENTO DE DADOS
ACESSO A DADOS
GOVERNANÇA E INTEGRAÇÃO DE DADOS
SEGURANÇA
OPERAÇÕES
CLOUD
PEDRA ANGULAR DA HORTONWORKS DATA PLATFORM
O YARN e o Hadoop Distributed File System (HDFS) são componentes básicos da Hortonworks Data Platform (HDP). Enquanto o HDFS forneça armazenamento econômico, escalável e tolerante a falhas para o seu big data lake, o YARN fornece arquitetura centralizada que permite processar várias cargas de trabalho simultaneamente. O YARN oferece o gerenciamento de recursos e a arquitetura plugável que permitem uma grande variedade de métodos de acesso a dados.
MAIS INFORMAÇÕES:
HDFS › YARN ›
YARN: DATA OPERATING SYSTEM
HDFS
Hadoop Distributed File System

Estrutura (5)

GERENCIAMENTO DE DADOS

ACESSO A DADOS

GOVERNANÇA E INTEGRAÇÃO DE DADOS

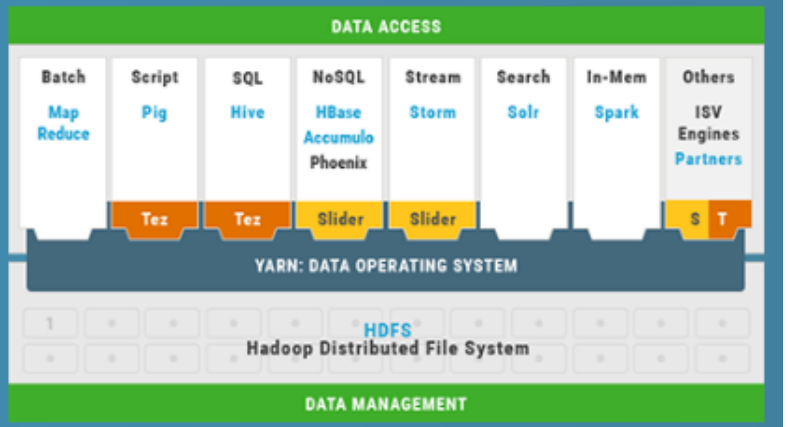
SEGURANÇA

OPERAÇÕES

CLOUD

MECANISMOS DE ANALÍTICA, PROCESSAMENTO E STREAMING DE DADOS PARA UMA VARIEDADE DE CARGAS DE TRABALHO

A Hortonworks Data Platform inclui uma gama versátil de mecanismos de processamento que permitem interagir com os mesmos dados de várias maneiras e ao mesmo tempo. Isso significa que os aplicativos podem interagir com os dados da melhor forma: do lote ao SQL interativo ou acesso de baixa latência com NoSQL. Casos de uso emergentes relacionados à ciência de dados, busca e streaming também são suportados pelo Apache Spark, Storm e Kafka.



MAIS INFORMAÇÕES:

Hive › Tez › Pig › Storm › Spark ›

HBase › Accumulo › Solr ›

Estrutura (6)

GERENCIAMENTO DE DADOS

ACESSO A DADOS

GOVERNANÇA E INTEGRAÇÃO DE DADOS

SEGURANÇA

OPERAÇÕES

CLOUD

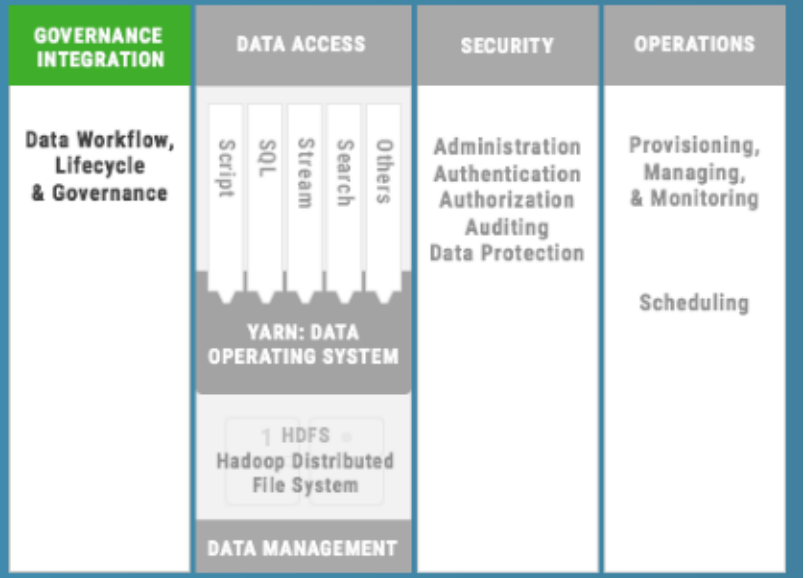
CARREGUE E GERENCIE DADOS DE ACORDO COM A POLÍTICA

A HDP amplia o acesso e gerenciamento de dados com ferramentas eficientes para governança e integração de dados. Elas oferecem uma estrutura simples, repetível e confiável para gerenciar o fluxo de dados de entrada e saída do Hadoop. Essa estrutura de controle, juntamente com um conjunto de ferramentas para facilitar e automatizar a aplicação de esquema ou metadados em fontes é crucial para o sucesso da integração do Hadoop em sua arquitetura moderna de dados.

A Hortonworks possui relações de engenharia com muitos fornecedores de gerenciamento de dados líderes para permitir que suas ferramentas funcionem e sejam integradas à HDP.

MAIS INFORMAÇÕES:

Atlas › Falcon › Oozie › Sqoop › Flume › Kafka ›



Estrutura (7)

GERENCIAMENTO DE DADOS

ACESSO A DADOS

GOVERNANÇA E INTEGRAÇÃO DE DADOS

SEGURANÇA

OPERAÇÕES

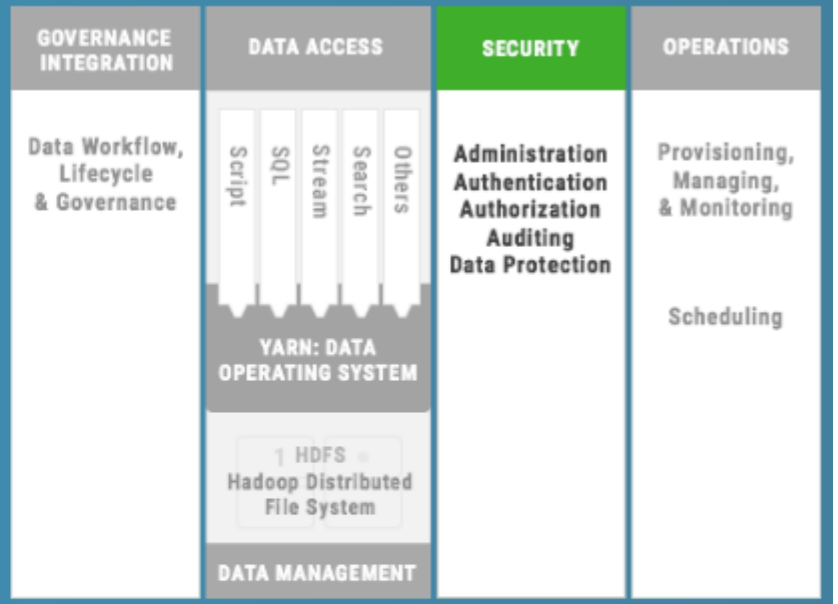
CLOUD

AUTENTICAÇÃO, AUTORIZAÇÃO E PROTEÇÃO DE DADOS

A segurança está impregnada e integrada na HDP em várias camadas. Recursos fundamentais para autenticação, autorização, prestação de contas e proteção de dados são empregados a fim de ajudar a proteger a HDP em todos esses requisitos importantes. De forma condizente com a abordagem em todos os recursos corporativos do Hadoop, a HDP também garante a possibilidade de integrar e ampliar suas soluções atuais de segurança para oferecer um guarda-chuva único, consistente e seguro sobre sua arquitetura moderna de dados.

MAIS INFORMAÇÕES:

Knox › Ranger ›



Estrutura (8)

GERENCIAMENTO DE DADOS

ACESSO A DADOS

GOVERNANÇA E INTEGRAÇÃO DE DADOS

SEGURANÇA

OPERAÇÕES

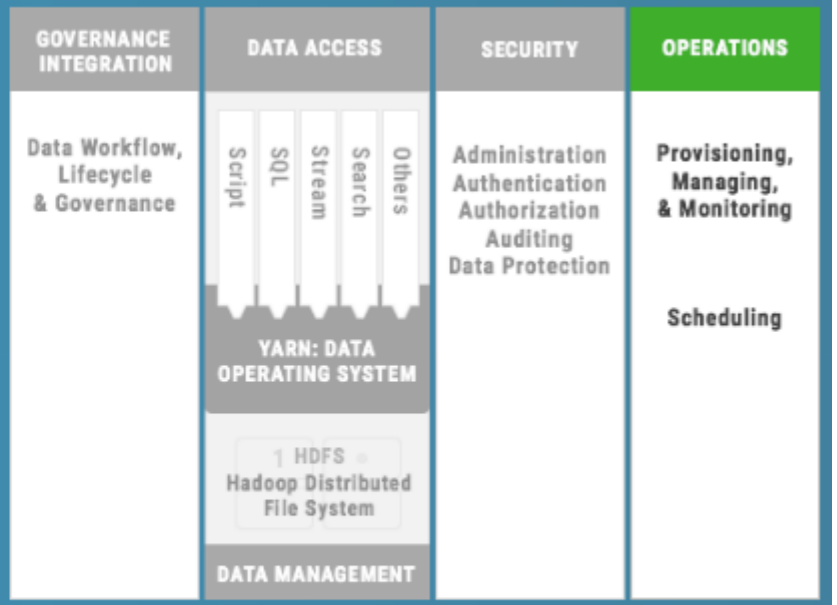
CLOUD

ACABE COM OS "ACHISMOS" USANDO HADOOP

Equipes de operações implementam, monitoram e gerenciam um cluster Hadoop em seus grandes ecossistemas de dados corporativos. O Apache Ambari simplifica essa experiência. O Ambari é uma plataforma de gerenciamento em código aberto para provisionar, gerenciar, monitorar e proteger a Hortonworks Data Platform. Permite que o Hadoop se adapte de forma impecável ao seu ambiente corporativo.

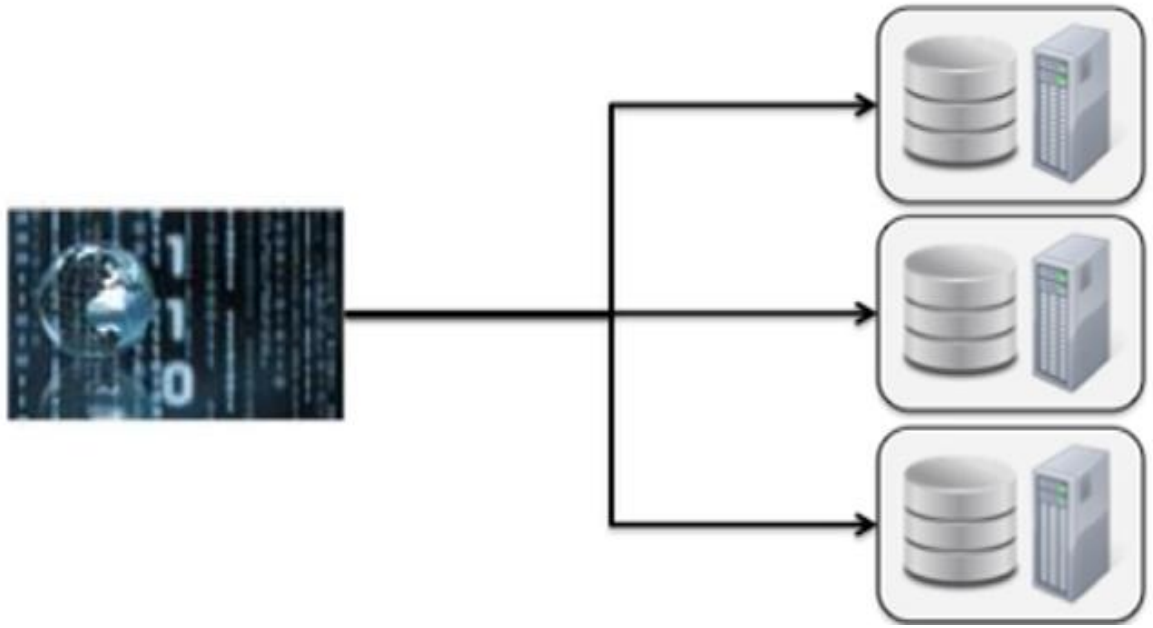
MAIS INFORMAÇÕES:

Ambari › Zookeeper ›



Funcionamento...

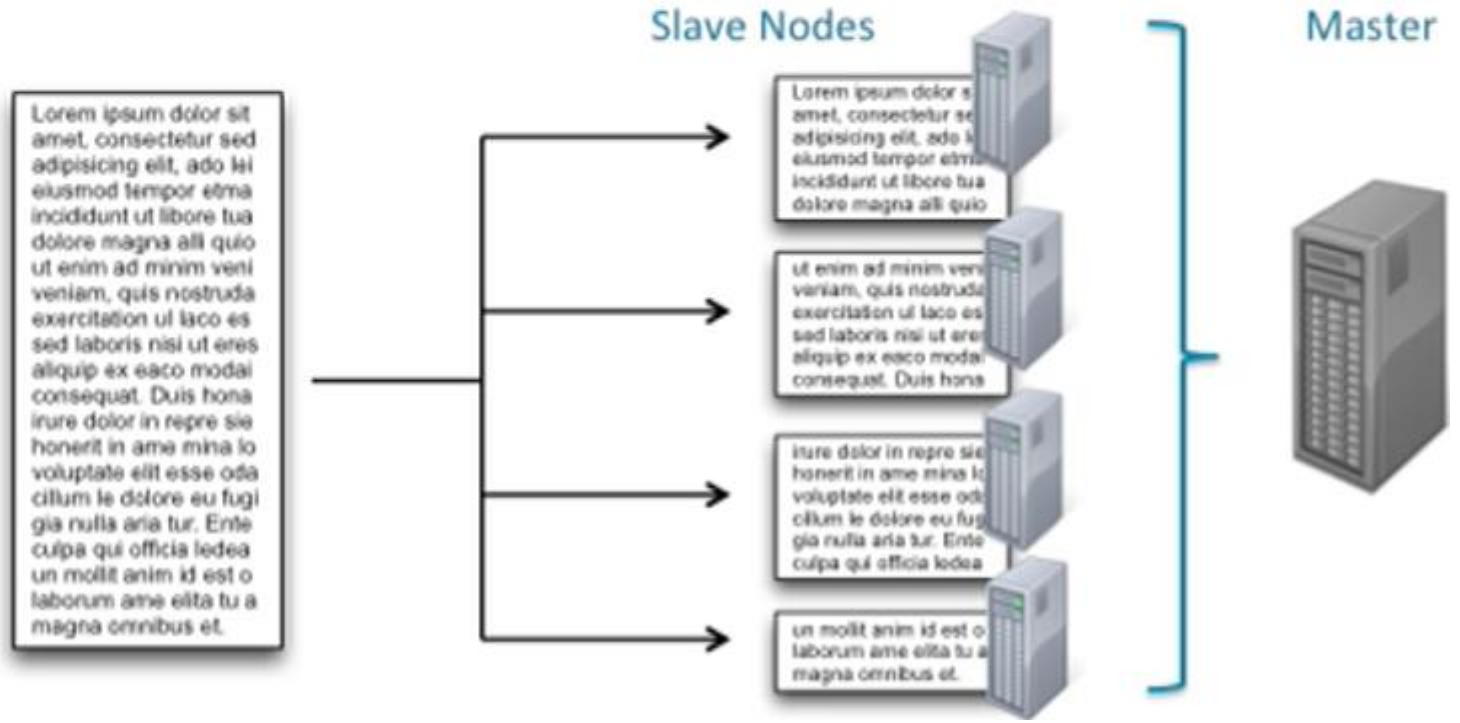
- **A radical new approach to distributed computing**
  - Distribute data when the data is stored
  - Run computation where the data is



Cloudera Essentials for Apache Hadoop | 1 of 6 | The Motivation for Hadoop Online Video Training

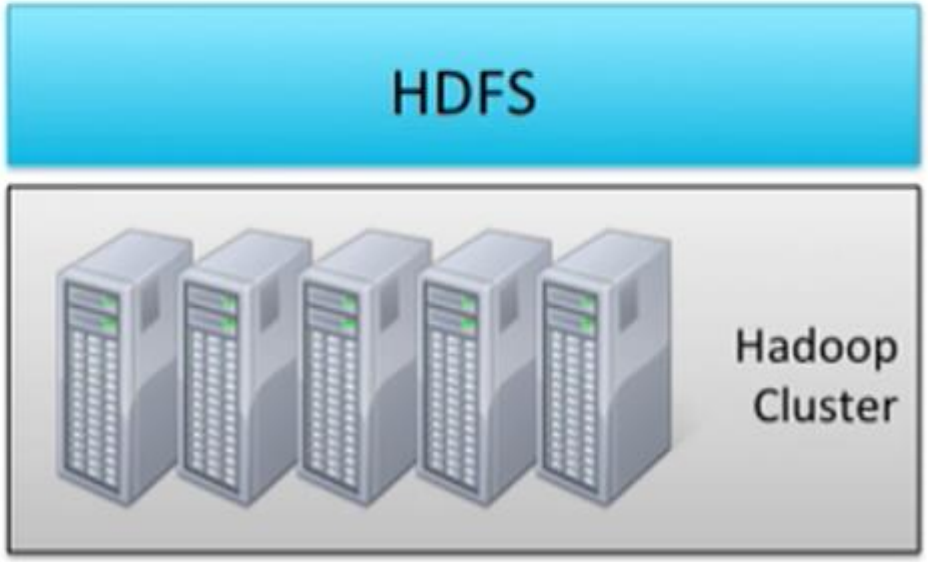
Hadoop: Very High-Level Overview

- **Data is split into "blocks" when loaded**
- **Map tasks typically work on a single block**
- **A master program manages tasks**



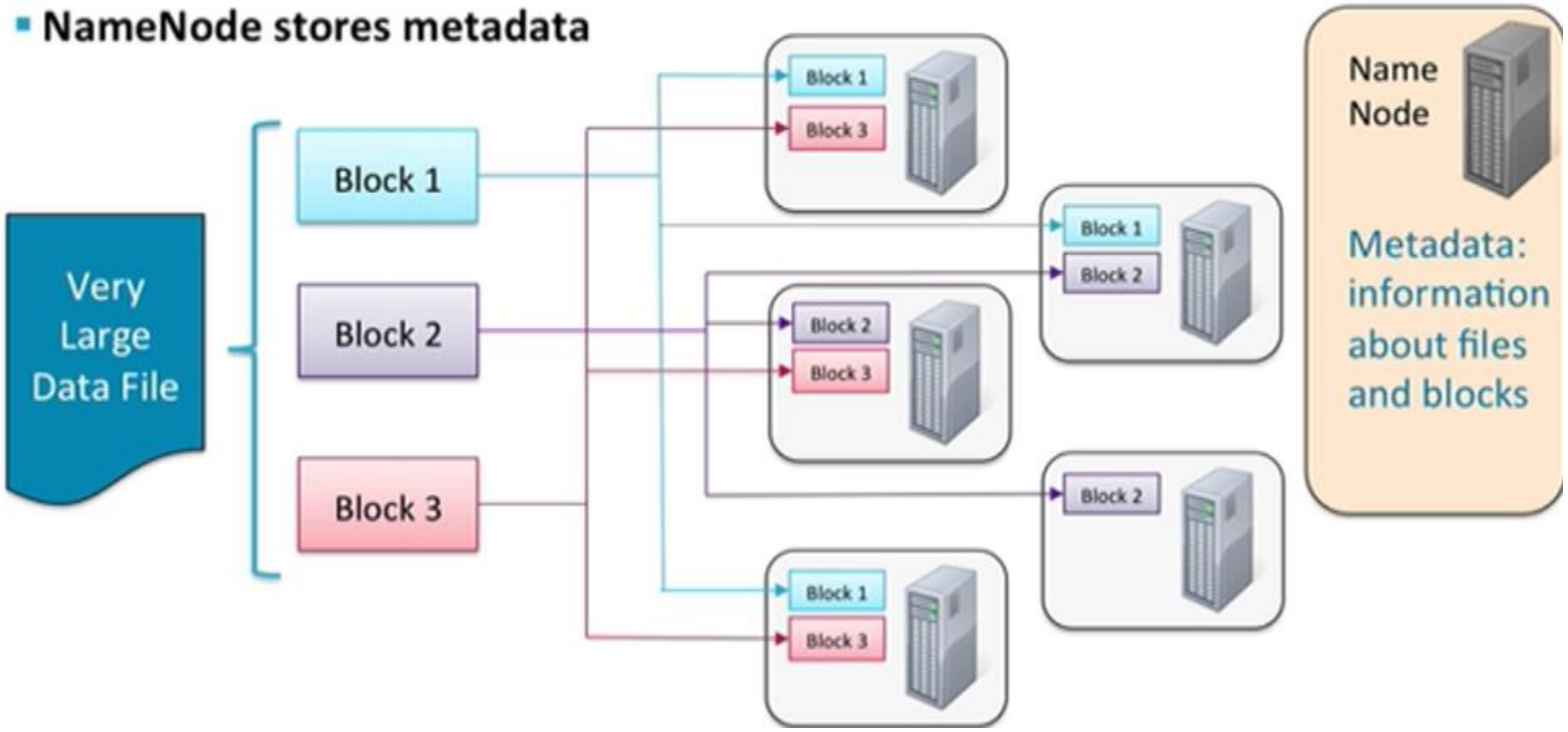
HDFS Basic Concepts

- **HDFS is a filesystem written in Java**
- **Sits on top of a native filesystem**
- **Storage for massive amounts of data**
  - Scalable
  - Fault tolerant
  - Supports efficient processing with MapReduce

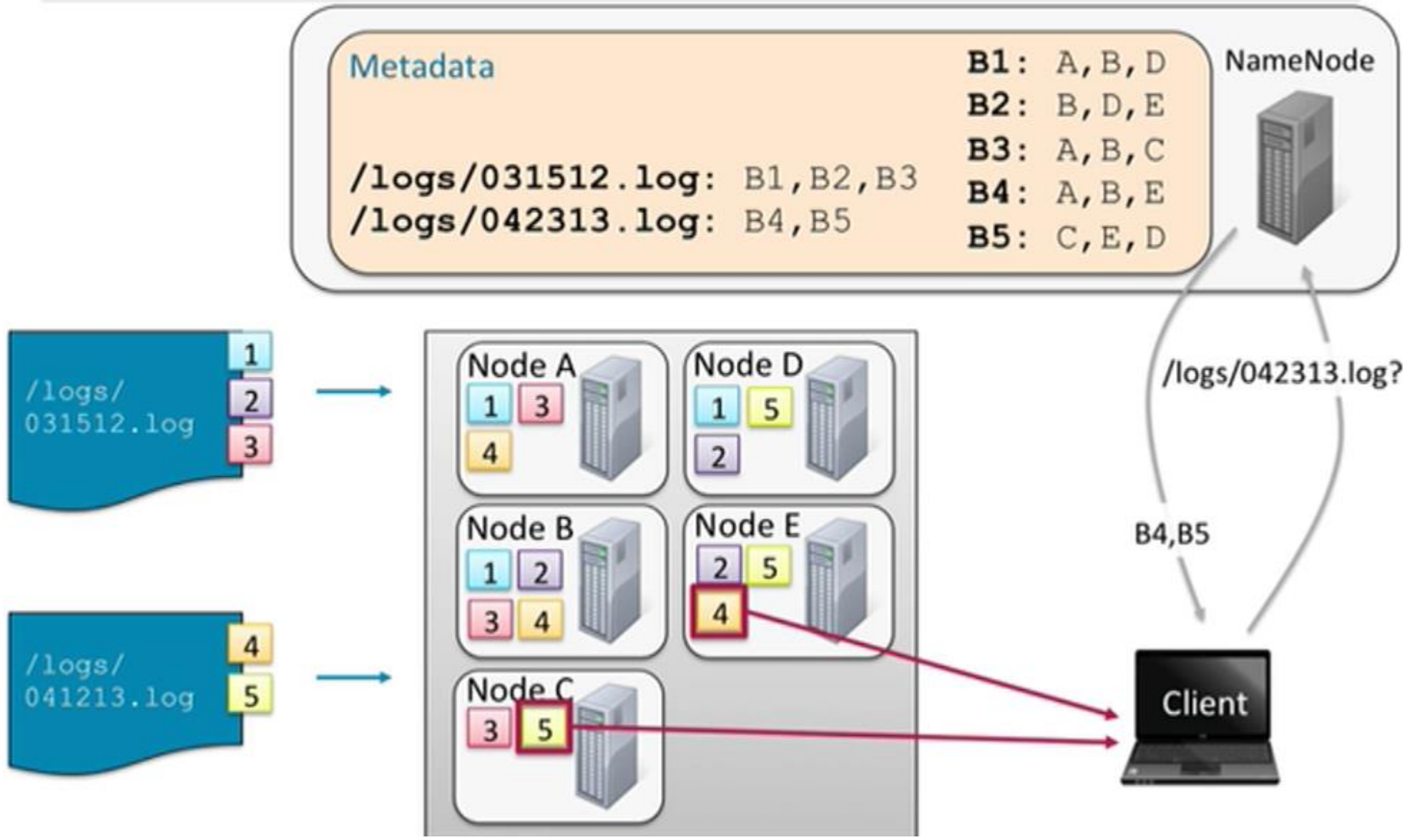


How Files are Stored (4)

- **Data files are split into blocks and distributed to data nodes**
- **Each block is replicated on multiple nodes (default 3x)**
- **NameNode stores metadata**



Example: Storing and Retrieving Files (4)



HDFS e MapReduce

HDFS

• HDFS performs best with a modest number of large files.

Millions, rather than billions, of files

Each file typically 100MB or more

• Files in HDFS are "write once"

Files can be replaced but not changed

• HDFS is optimized for large, streaming reads of files

Rather than random reads

MapReduce

• The Mapper

Each Map task (typically) operates on a single HDFS block

Map tasks run on the node where the block is stored (usually)

• Shuffle & Sort

Sorts and consolidates intermediate data from all mappers

Happens after all Map tasks are complete and before Reduce tasks start

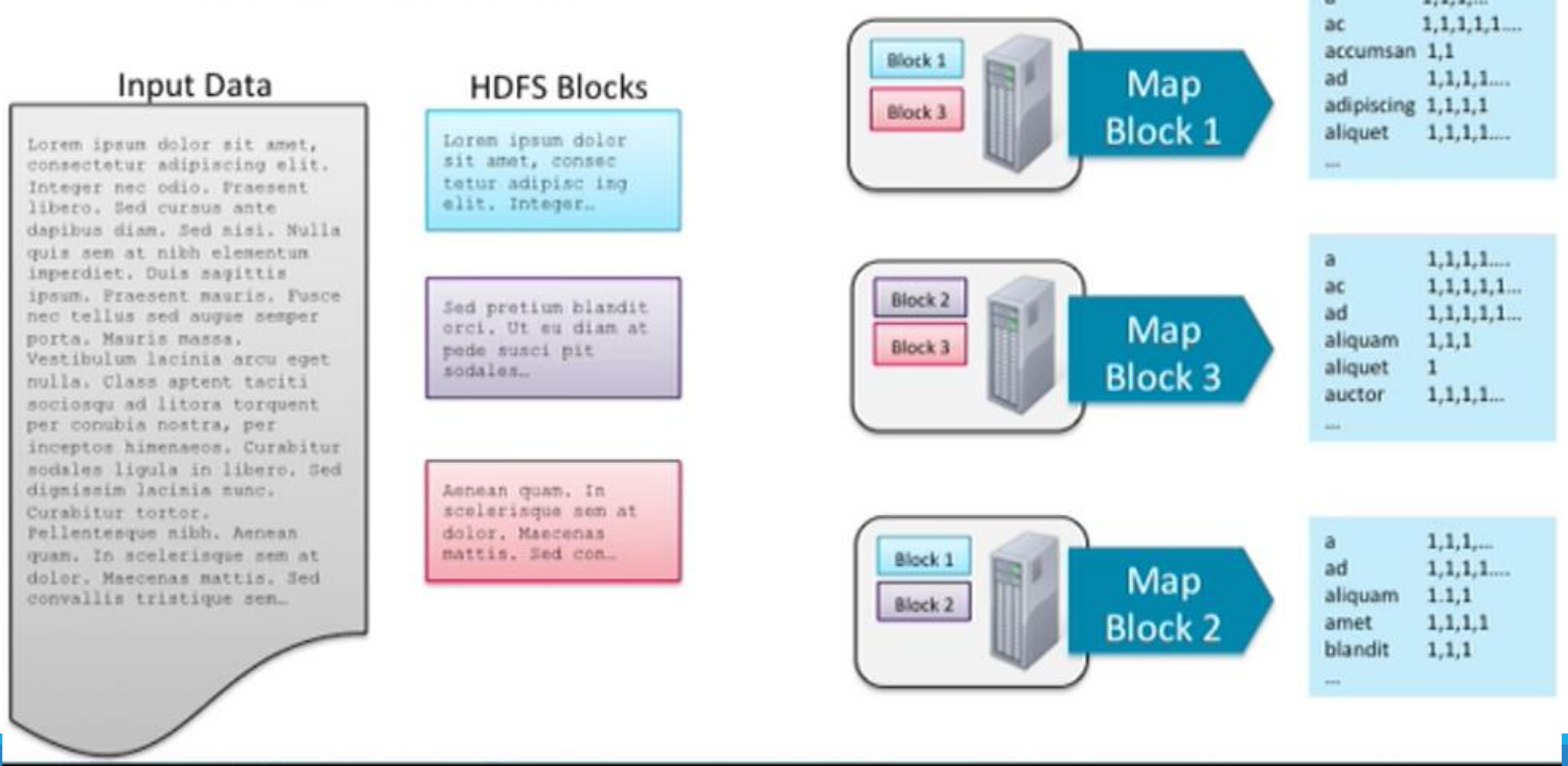
• The Reducer

Operates on sorted/shuffled intermediate data (Map task output)

Produces final output [Cloudera]

Mappers Run in Parallel

- **Hadoop runs Map tasks on the slave node storing the block (when possible)**
  - many Mappers can run in parallel
  - minimizes network traffic



Example: Shuffle & Sort

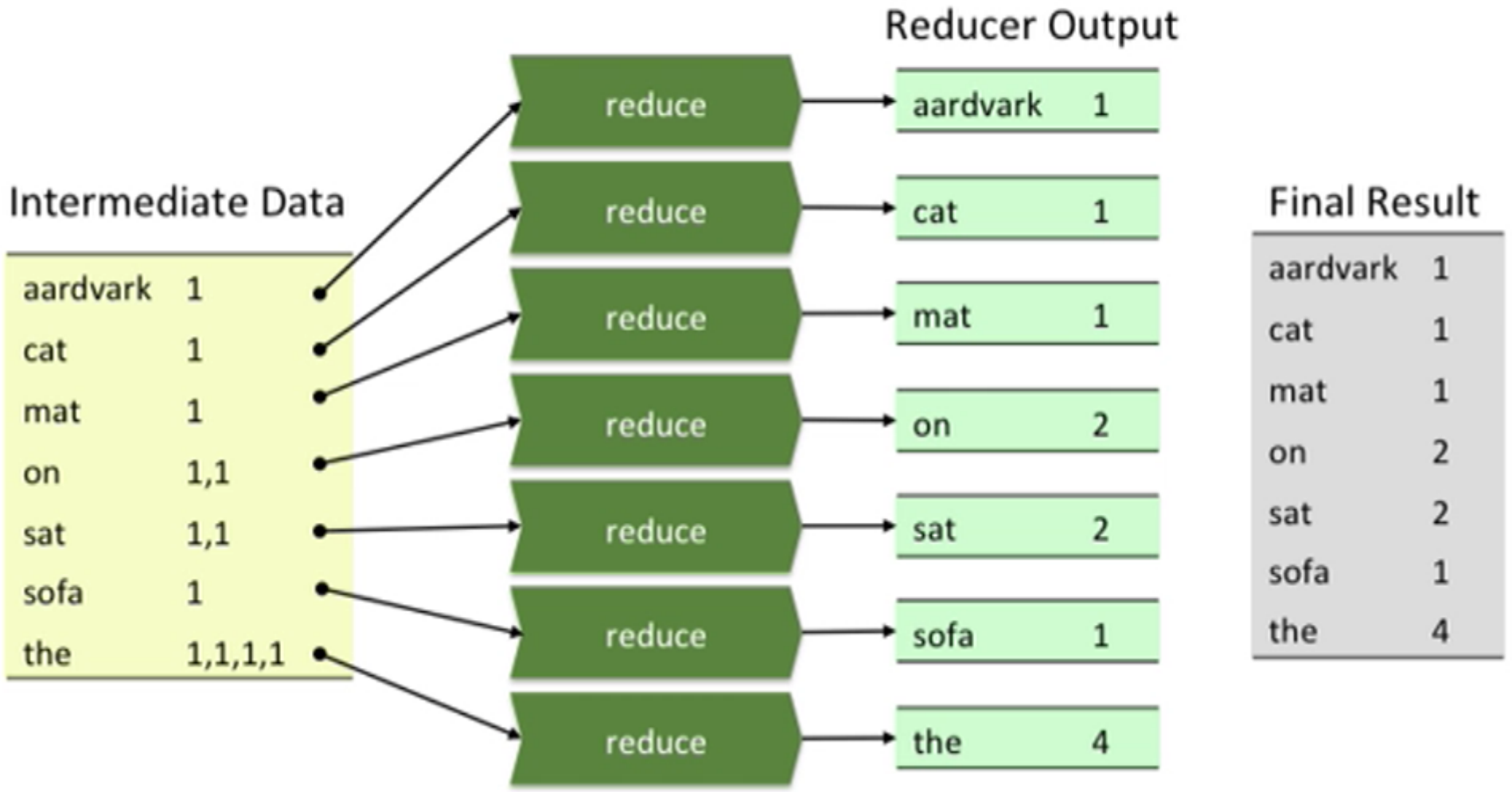
Mapper Output

The	1
cat	1
sat	1
on	1
the	1
mat	1
The	1
aardvark	1
sat	1
on	1
the	1
sofa	1

Intermediate Data

aardvark	1
cat	1,1
mat	1
on	1,1
sat	1,1
sofa	1
the	1,1,1,1

Example: SumReducer



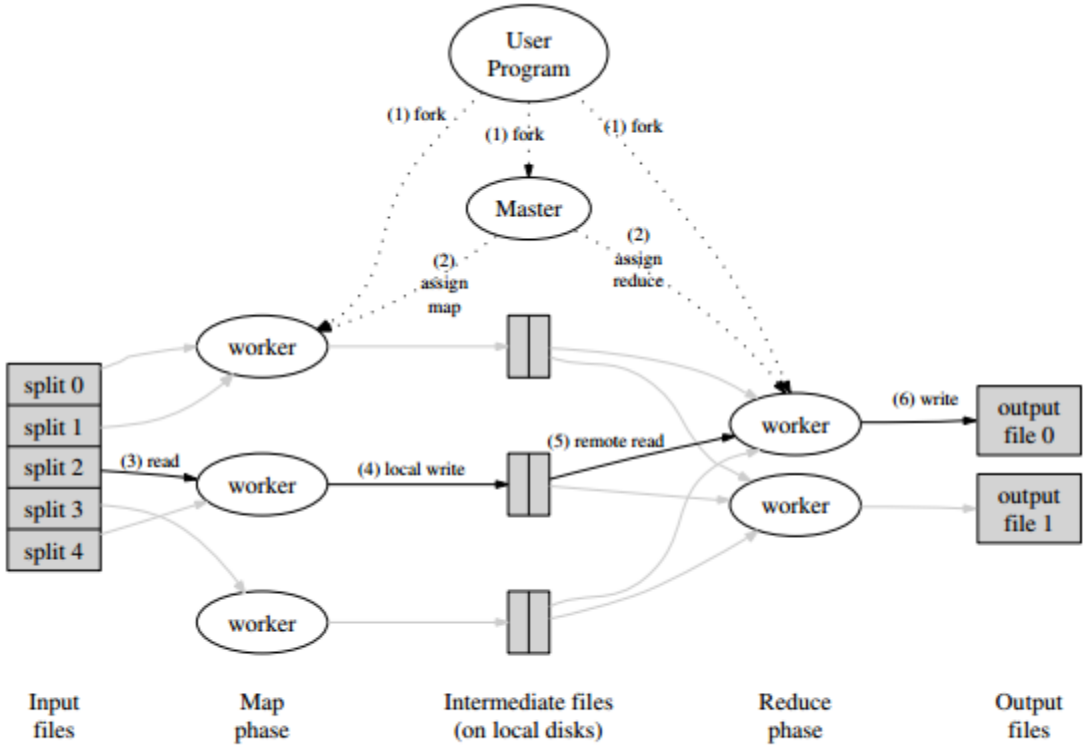
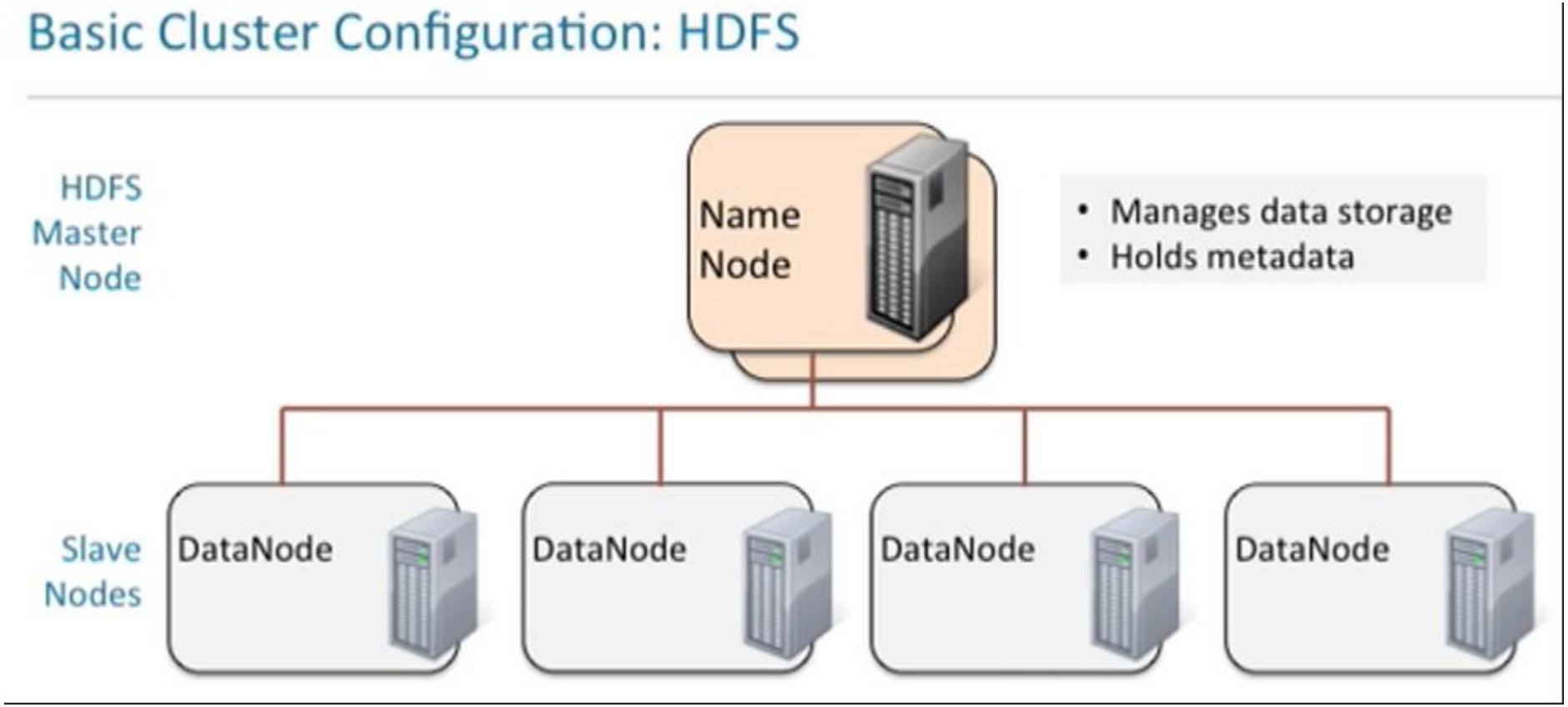


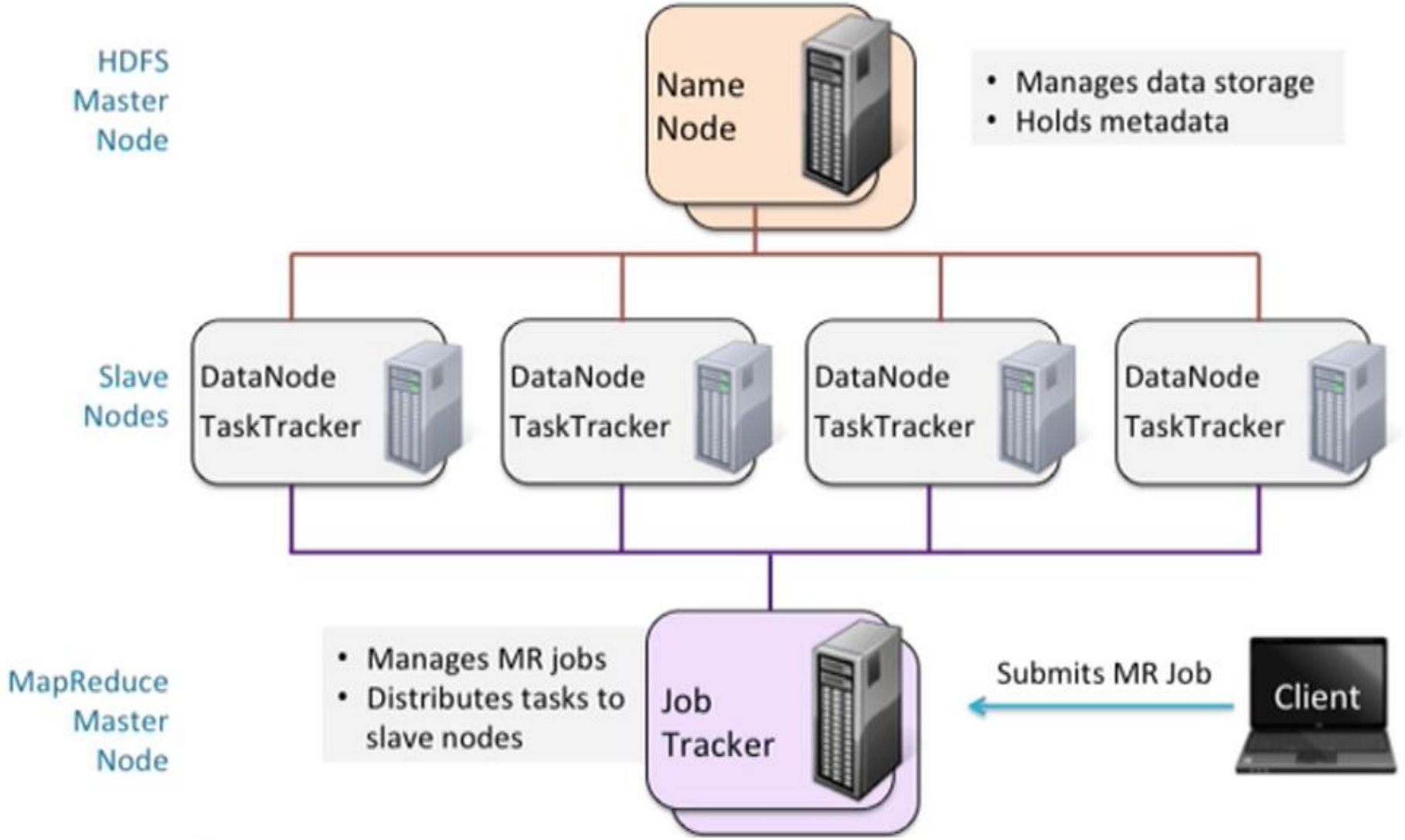
Figure 1: Execution overview

Imagem do artigo original.

Basic Cluster Configuration: HDFS



Basic Cluster Configuration: MapReduce



Name node e job tracker podem ter um host backup para garantir alta disponibilidade.

Inserindo dados

API web ou CLI do HDFS.

Sqoop: Conexão com bancos de dados relacionais;

Flume: Conexão com fluxo de dados em tempo real, por exemplo, arquivos de log.

O FIWARE Cygnus é uma implementação do Apache Flume.

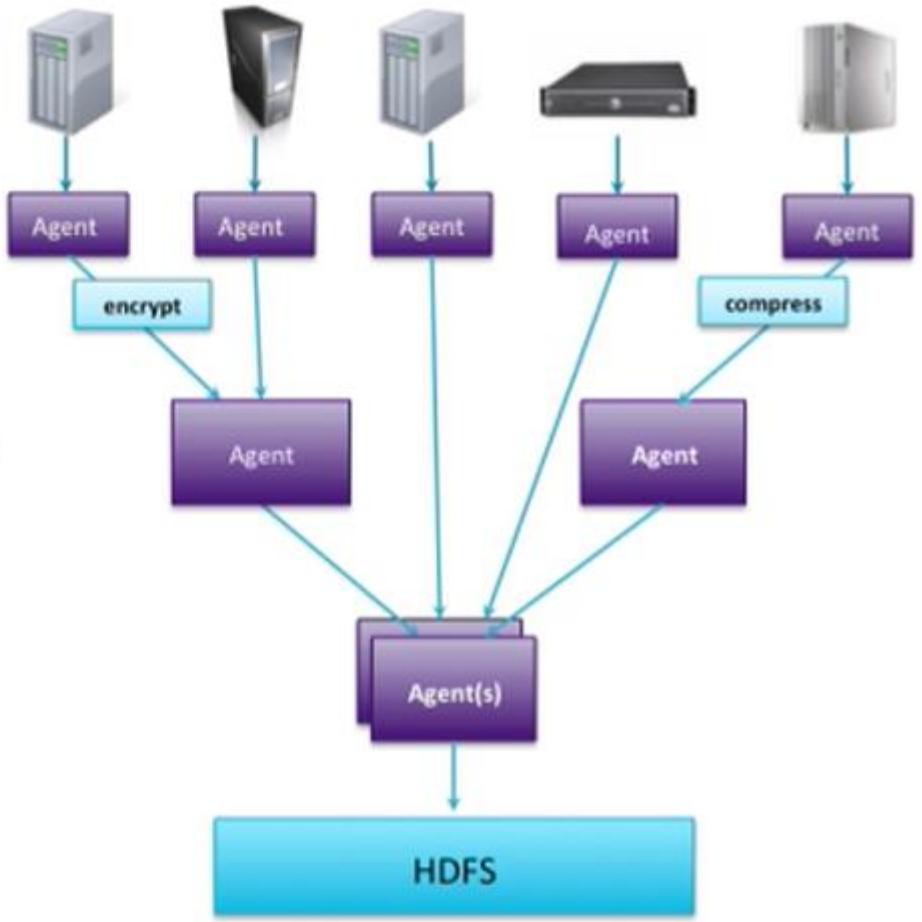
Getting Data In and Out of HDFS

- **Hadoop**
  - Copies data between client (local) and HDFS (cluster)
  - API or command line
- **Ecosystem Projects**
  - Flume
    - Collects data from network sources (e.g. websites, system logs)
  - Sqoop
    - Transfers data between HDFS and RDBMS
- **Business Intelligence Tools**

RDBMS: Relational Database Management System

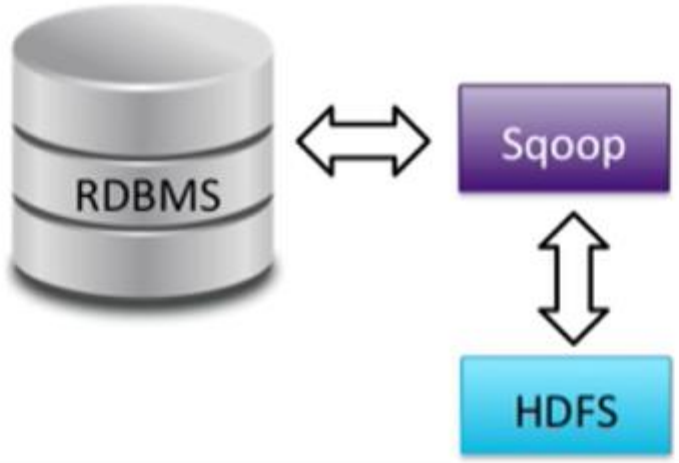
Flume: High-Level Overview

- Collect data as it is produced
  - Files, syslogs, stdout or custom source
- Process in place
  - e.g., encrypt, compress
- Pre-process data before storing
  - e.g., transform, scrub, enrich
- Write in parallel
  - Scalable throughput
- Store in any format
  - Text, compressed, binary, or custom sink

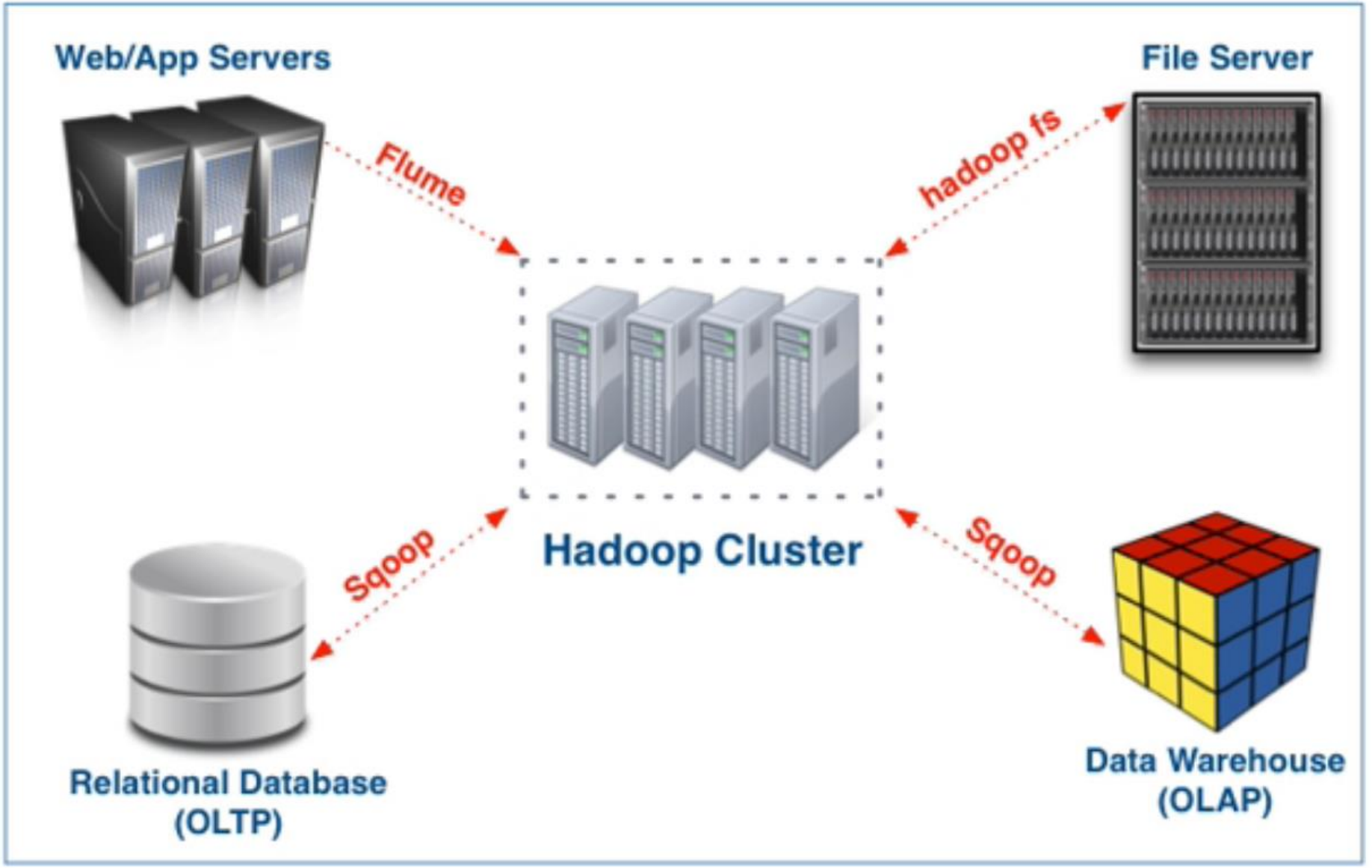


Sqoop: Exchanging Data With RDBMSs

- **Sqoop: SQL to Hadoop**
  - Transfers data between RDBMS and HDFS
  - Uses a command-line tool or application connector
  - Allows incremental imports



Data Center Integration



Pig and Hive

Improving programmability: Working directly with Java APIs can be tedious and error prone. It also restricts usage of Hadoop to Java programmers. Hadoop offers two solutions for making Hadoop programming easier.

Pig is a programming language that simplifies the common tasks of working with Hadoop: loading data, expressing transformations on the data, and storing the final results. Pig's built-in operations can make sense of semi-structured data, such as log files, and the language is extensible using Java to add support for custom data types and transformations.

Hive enables Hadoop to operate as a data warehouse. It superimposes structure on data in HDFS and then permits queries over the data using a familiar SQL-like syntax. As with Pig, Hive's core capabilities are extensible.

Open source Apache projects

- Hive initially developed at Facebook
- Pig Initially developed at Yahoo!

HBASE

Welcome to Apache HBase™

Apache HBase™ is the Hadoop non-relational (NoSQL) database, a distributed, scalable, big data store.

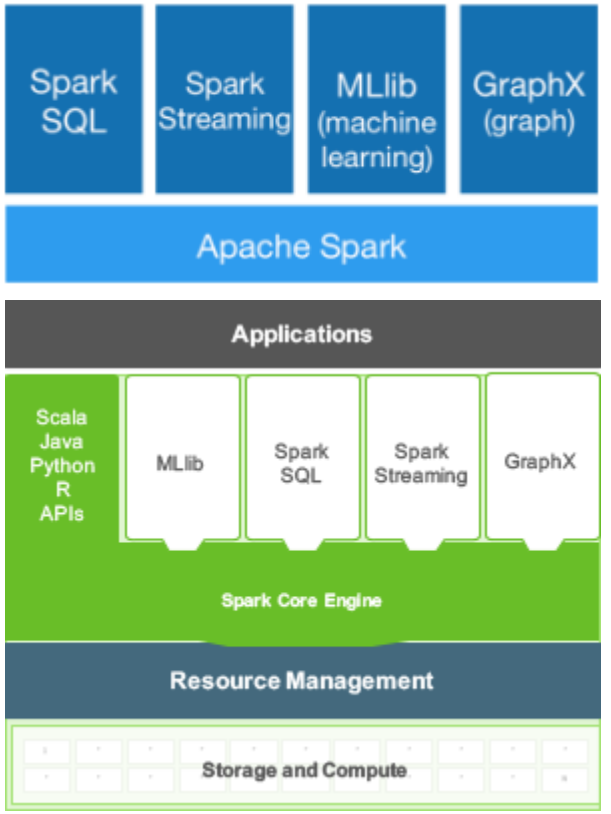
Use Apache HBase™ when you need random, realtime read/write access to your Big Data. This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, non-relational database modeled after Google's Bigtable: A Distributed Storage System for Structured Data by Chang et al.

Apache Spark

Uma “engine” para processamento de dados em larga escala, sendo capaz de processar ETL (Extract-Transform-Load), machine learning e workloads de ciência de dados em memória.

Possui uma série de bibliotecas que podem ser combinadas em uma mesma aplicação.

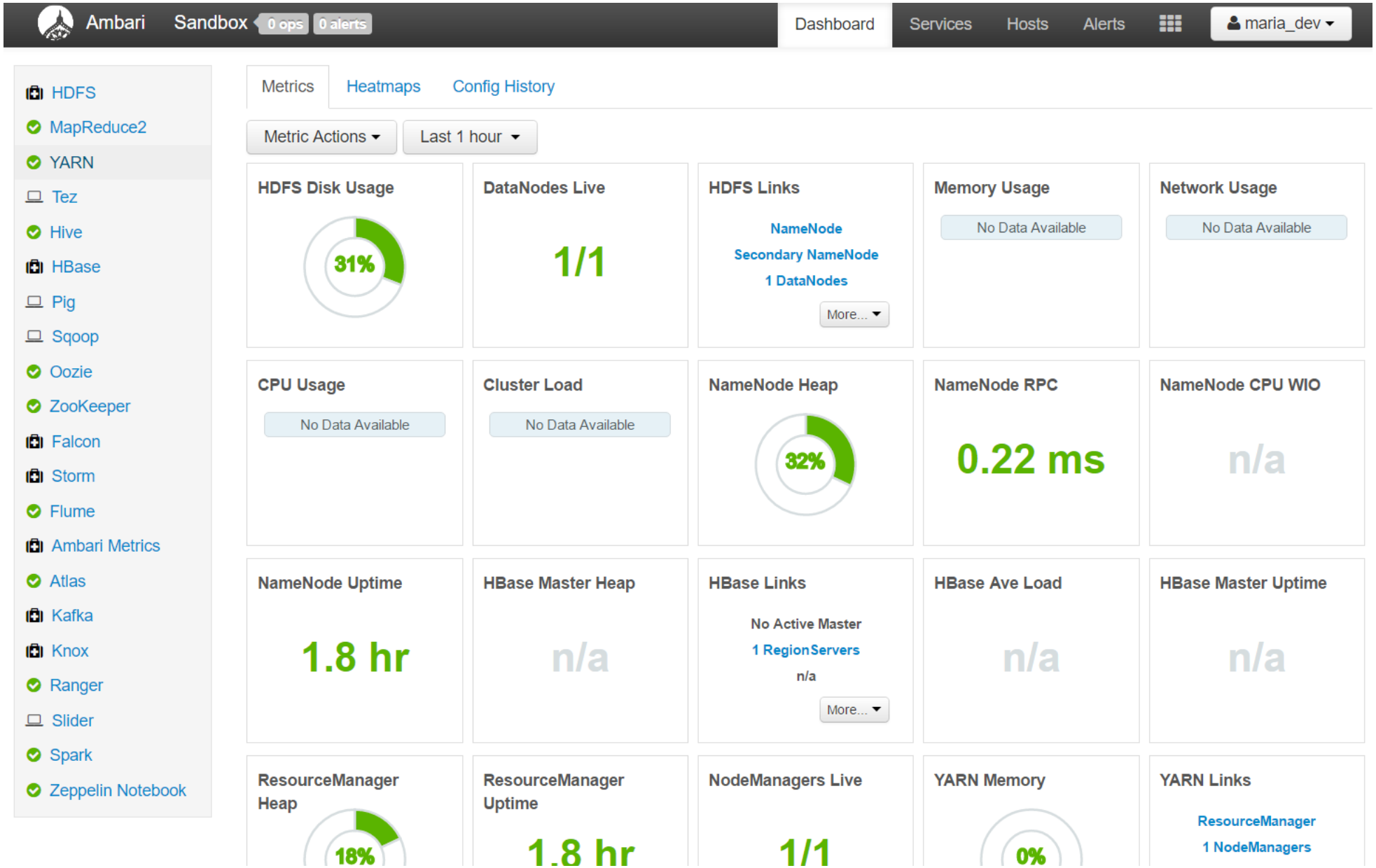
- **MLlib** is Apache Spark's scalable machine learning library.
- **Spark SQL** is Apache Spark's module for working with structured data.
- **Spark Streaming** makes it easy to build scalable fault-tolerant streaming applications.
- **GraphX** is Apache Spark's API for graphs and graph-parallel computation.



Ambari

APACHE AMBARI

A completely open source management platform for provisioning, managing, monitoring and securing Apache Hadoop clusters. Apache Ambari takes the guesswork out of operating Hadoop.


Ambari
Sandbox
0 ops
0 alerts
Dashboard
Services
Hosts
Alerts
maria_dev
HDFS
MapReduce2
YARN
Tez
Hive
HBase
Pig
Sqoop
Oozie
ZooKeeper
Falcon
Storm
Flume
Ambari Metrics
Atlas
Kafka
Knox
Ranger
Slider
Spark
Zeppelin Notebook
Metrics
Heatmaps
Config History
Metric Actions
Last 1 hour
HDFS Disk Usage
31%
DataNodes Live
1/1
HDFS Links
NameNode
Secondary NameNode
1 DataNodes
More...
Memory Usage
No Data Available
Network Usage
No Data Available
CPU Usage
No Data Available
Cluster Load
No Data Available
NameNode Heap
32%
NameNode RPC
0.22 ms
NameNode CPU WIO
n/a
NameNode Uptime
1.8 hr
HBase Master Heap
n/a
HBase Links
No Active Master
1 RegionServers
n/a
More...
HBase Ave Load
n/a
HBase Master Uptime
n/a
ResourceManager Heap
18%
ResourceManager Uptime
1.8 hr
NodeManagers Live
1/1
YARN Memory
0%
YARN Links
ResourceManager
1 NodeManagers

Testando - SandBoxes

As distribuições disponibilizam “Single Node Clusters” implementados em máquinas virtuais prontas:

- http://br.hortonworks.com/products/sandbox/
  - No Microsoft Azure: http://br.hortonworks.com/hadoop-tutorial/deploying-hortonworks-sandbox-on-microsoft-azure/
  - Tutorial: http://br.hortonworks.com/hadoop-tutorial/learning-the-ropes-of-the-hortonworks-sandbox/
- http://www.cloudera.com/downloads.html
- https://www.mapr.com/products/mapr-sandbox-hadoop/download

Testando - Hadoop Puro

Hadoop: Setting up a Single Node Cluster.

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html

Outros links úteis:

Setup Virtual Hadoop Cluster under Ubuntu with VirtualBox:
http://cs.smith.edu/dftwiki/index.php/Setup_Virtual_Hadoop_Cluster_under_Ubuntu_with_VirtualBox

How to create Hadoop cluster on Virtual Machines Running on same Laptop/Desktop:
https://hadoopmagic.wordpress.com/2013/07/13/how-to-create-hadoop-cluster-on-virtual-machines-running-on-same-laptopdesktop/

Running Hadoop on Ubuntu Linux (Single-Node Cluster): http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/

Running Hadoop on Ubuntu Linux (Multi-Node Cluster): http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/

Creating a virtualized fully-distributed Hadoop cluster using Linux Containers:
https://ofirm.wordpress.com/2014/01/05/creating-a-virtualized-fully-distributed-hadoop-cluster-using-linux-containers/

Alternativas OpenSource

- HPCC: HPCC Systems (High Performance Computing Cluster) is an open source, massive parallel-processing computing platform for big data processing and analytics.
  https://hpccsystems.com/

- Spark é tido como um ecossistema em si. É possível executá-lo em um cluster com Apache Mesos no lugar do YARN.
  http://spark.apache.org/docs/latest/running-on-mesos.html

Cosmos: Hadoop no FIWARE

FIWARE's BigData Analysis Generic Enabler (GE) que oferece um Hadoop As A Service no Fiware.

O Orion não tem armazenamento persistente. O Cosmos adiciona armazenamento persistente à arquitetura. O Cygnus (baseado no Apache Flume) permite a conexão do Orion com o Cosmos.

- Página no catálogo de GEs do Fiware: https://catalogue.fiware.org/enablers/bigdata-analysis-cosmos
- Documentação: http://fiware-cosmos.readthedocs.io/
- GitHub: https://github.com/telefonicaid/fiware-cosmos
- Curso na FIWARE Academy: https://edu.fiware.org/course/view.php?id=69