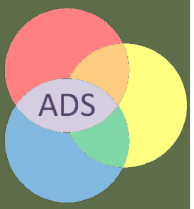
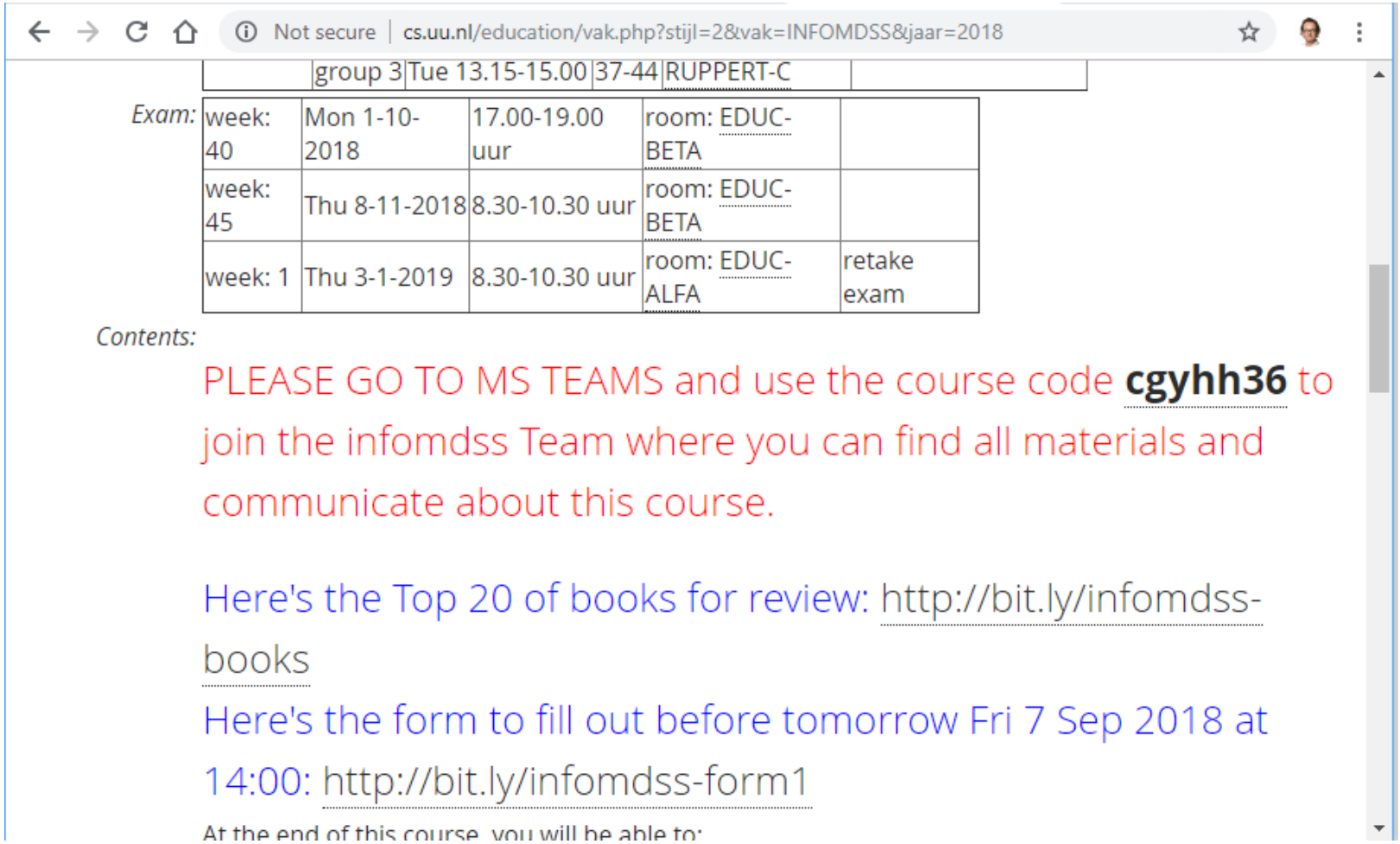
Data Science & Society

Lecture 02:
Distributed Computing with Hadoop

INFOMDSS 2018 :: Dr. Marco Spruit



Bootstrapping this course...



Not secure | cs.uu.nl/education/vak.php?stijl=2&vak=INFOMDSS&jaar=2018

	group 3	Tue 13.15-15.00	37-44	RUPPERT-C	

Exam:

week: 40	Mon 1-10-2018	17.00-19.00 uur	room: EDUC-BETA	
week: 45	Thu 8-11-2018	8.30-10.30 uur	room: EDUC-BETA	
week: 1	Thu 3-1-2019	8.30-10.30 uur	room: EDUC-ALFA	retake exam

Contents:

PLEASE GO TO MS TEAMS and use the course code **cgyhh36** to join the infomdss Team where you can find all materials and communicate about this course.

Here's the Top 20 of books for review: http://bit.ly/infomdss-books

Here's the form to fill out before tomorrow Fri 7 Sep 2018 at 14:00: http://bit.ly/infomdss-form1

At the end of this course, you will be able to:

Agenda

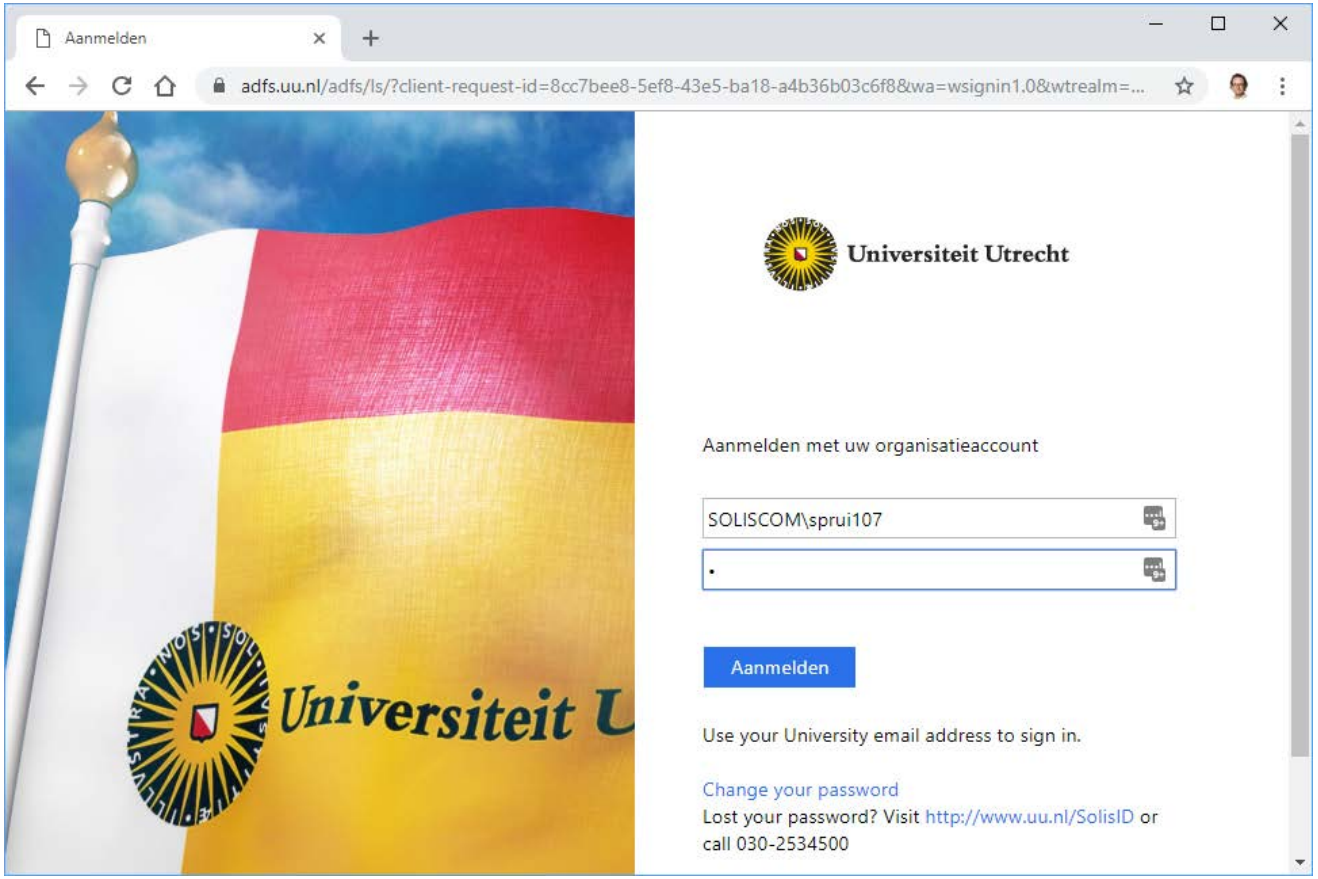
- Azure in Context

- Hadoop Beginnings...
- Hadoop in a Nutshell
- The Apache Hadoop Framework
  - HDFS
  - YARN
  - Zoo/Ecosystem

→ *Book review...*

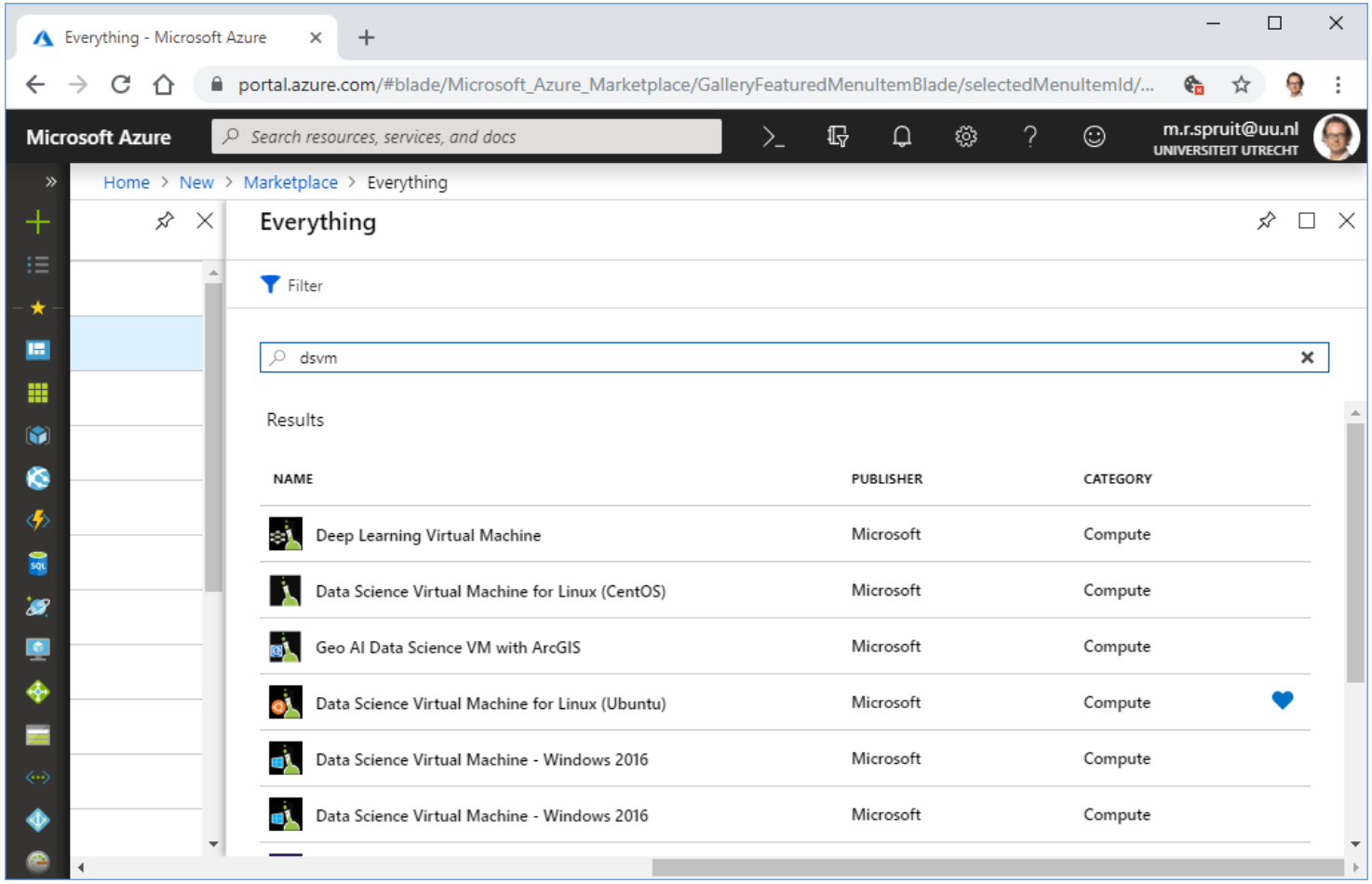
MS Azure

› http://portal.azure.com



Reuse your knowledge later....

› Create a Resource, type DSVM to Select a preconfigured VM



Create a personal account for your own projects

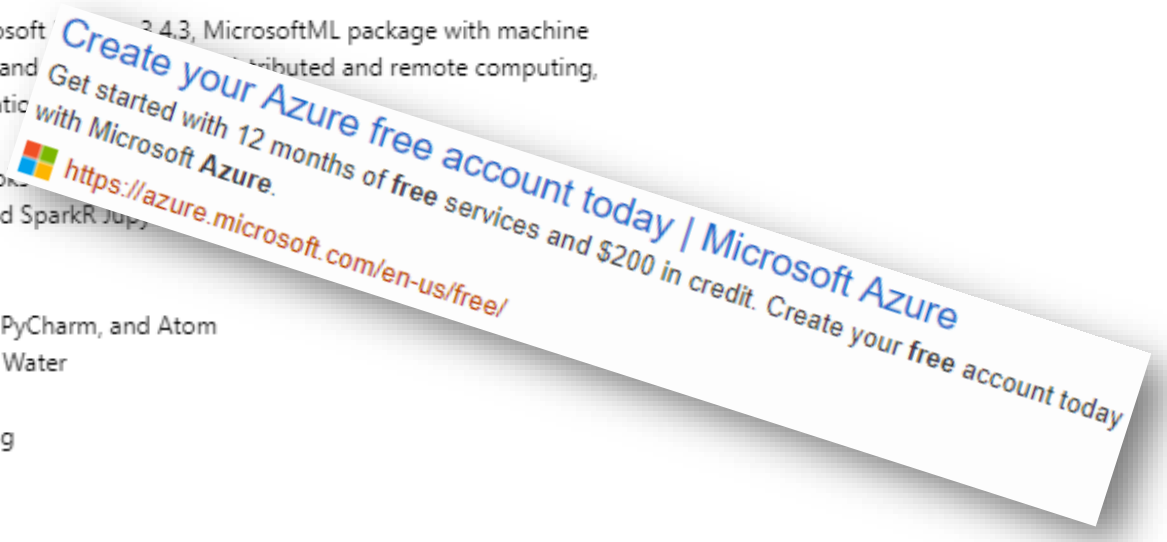
Data Science Virtual Machine for Linux (Ubuntu)

Microsoft

The Data Science Virtual Machine for Linux is an Ubuntu-based virtual machine image that makes it easy to get started with deep learning on Azure. The Microsoft Cognitive Toolkit, TensorFlow, MXNet, Caffe, Caffe2, Chainer, NVIDIA DIGITS, Deep Water, Keras, Theano, Torch, and PyTorch are built, installed, and configured so they are ready to run immediately. The NVIDIA driver, CUDA 9, and cuDNN 7 are also included. All frameworks are the GPU versions but work on the CPU as well. Many sample Jupyter notebooks are included. TensorFlow Serving, MXNet Model Server, and TensorRT are included to test inferencing.

The Data Science Virtual Machine for Linux also contains popular tools for data science and development activities, including:

- Microsoft R Server 9.3 with Microsoft … 3.4.3, MicrosoftML package with machine learning algorithms, RevoScaleR and … ributed and remote computing, and R and Python Operationalizatio…
- Anaconda Python 2.7 and 3.5
- JupyterHub with sample notebook…
- Spark local 2.3.1 with PySpark and SparkR Jup…
- Single node local Hadoop
- Azure command-line interface
- Visual Studio Code, IntelliJ IDEA, PyCharm, and Atom
- H2O, Deep Water, and Sparkling Water
- Julia
- Vowpal Wabbit for online learning
- xgboost for gradient boosting
- SQL Server 2017
- Intel Math Kernel Library



Amazon EC2

Resizable compute capacity in the Cloud.

Learn More »

750 hours per month of Linux, RHEL, or SLES t2.micro instance usage

750 hours per month of Windows t2.micro instance usage

For example, run 1 instance x 1 month or 2 instances x half a month

Expires 12 months after sign-up.

Amazon S3

Secure, durable, and scalable object storage infrastructure.

Learn More »

5 GB of Standard Storage

20,000 Get Requests

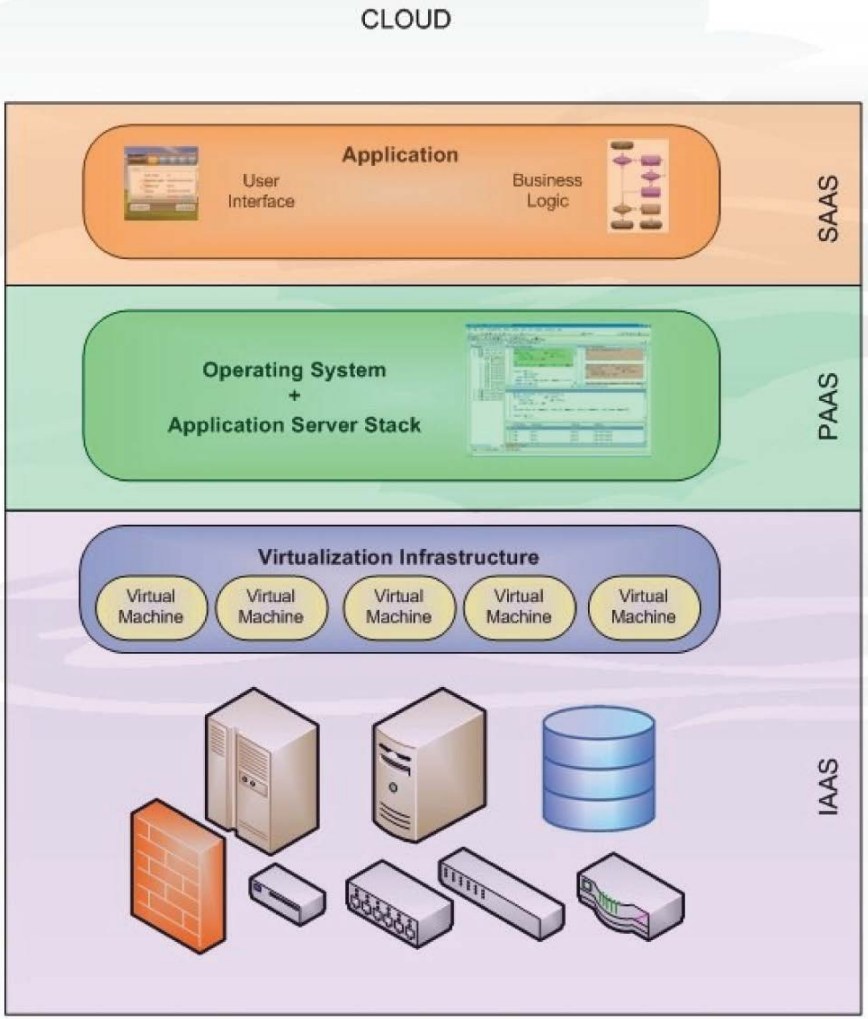
2,000 Put Requests

Expires 12 months after sign-up.

d.d. 30 March 2017: https://aws.amazon.com/free

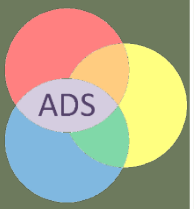
Cloud Computing: An Ontology

- Further reading:
  - Abdat,N., Spruit,M., & Bos,M. (2011). Software as a Service and the Pricing Strategy for Vendors. In Strader,T. (Ed.), *Digital Product Management, Technology and Practice: Interdisciplinary Perspectives, Advances in E-Business Research (AEBR) Book Series* (pp. 154–192). IGI Global. [pdf]



Apache Hadoop

Beginnings & In a Nutshell



What is Hadoop?

- Apache Hadoop is an open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware
- Created by Doug Cutting and Mike Cafarella in 2005
- Named the project after son's toy elephant

https://en.wikipedia.org/wiki/Apache_Hadoop

› Event 1 in the Hadoop origins timeline:

Google **Research Publications** Google Research Home

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

Abstract

We have designed and implemented the Google File System, a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients.

While sharing many of the same goals as previous distributed file systems, our design has been driven by observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system assumptions. This has led us to reexamine traditional choices and explore radically different design points.

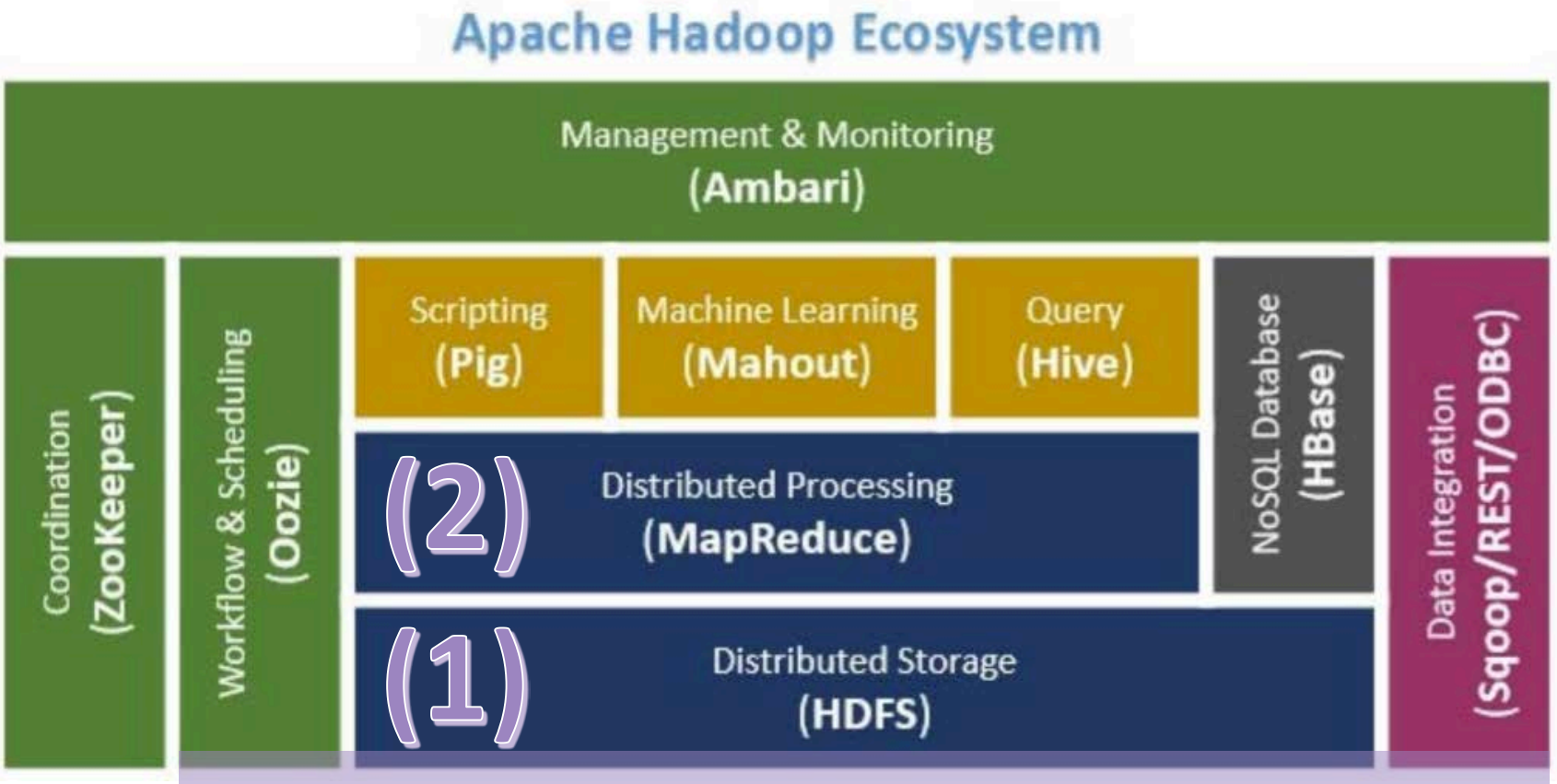
The file system has successfully met our storage needs. It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets. The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients.

In this paper, we present file system interface extensions designed to support distributed applications, discuss many aspects of our design, and report measurements from both micro-benchmarks and real world use.

Appeared in:
19th ACM Symposium on Operating Systems Principles,
Lake George, NY, October, 2003.

Download: PDF Version

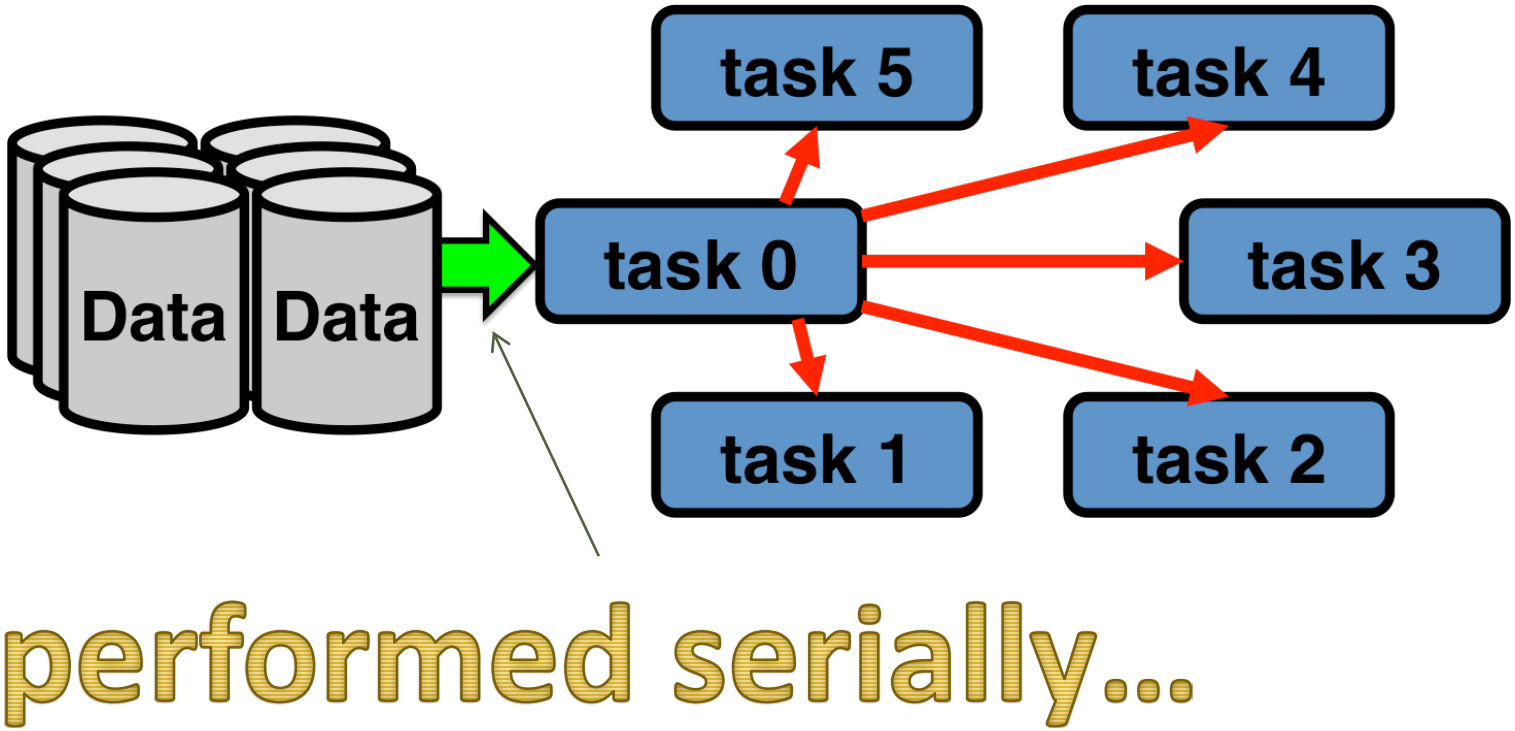
See Drive's "Literature\Required"



(1) + (2) = Distributed Computing in Hadoop

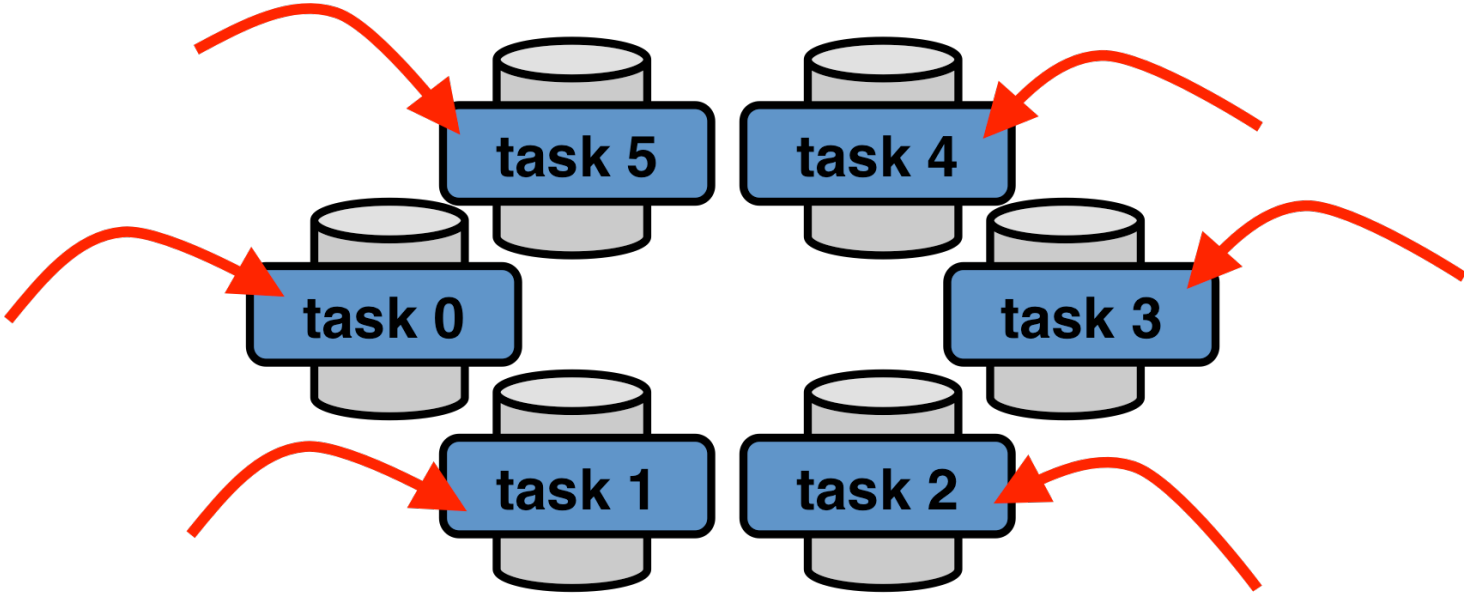
Comparing Map-Reduce to Traditional Parallelism

› map-reduce brings **compute to the data** in contrast to traditional parallelism, which brings data to the compute resources



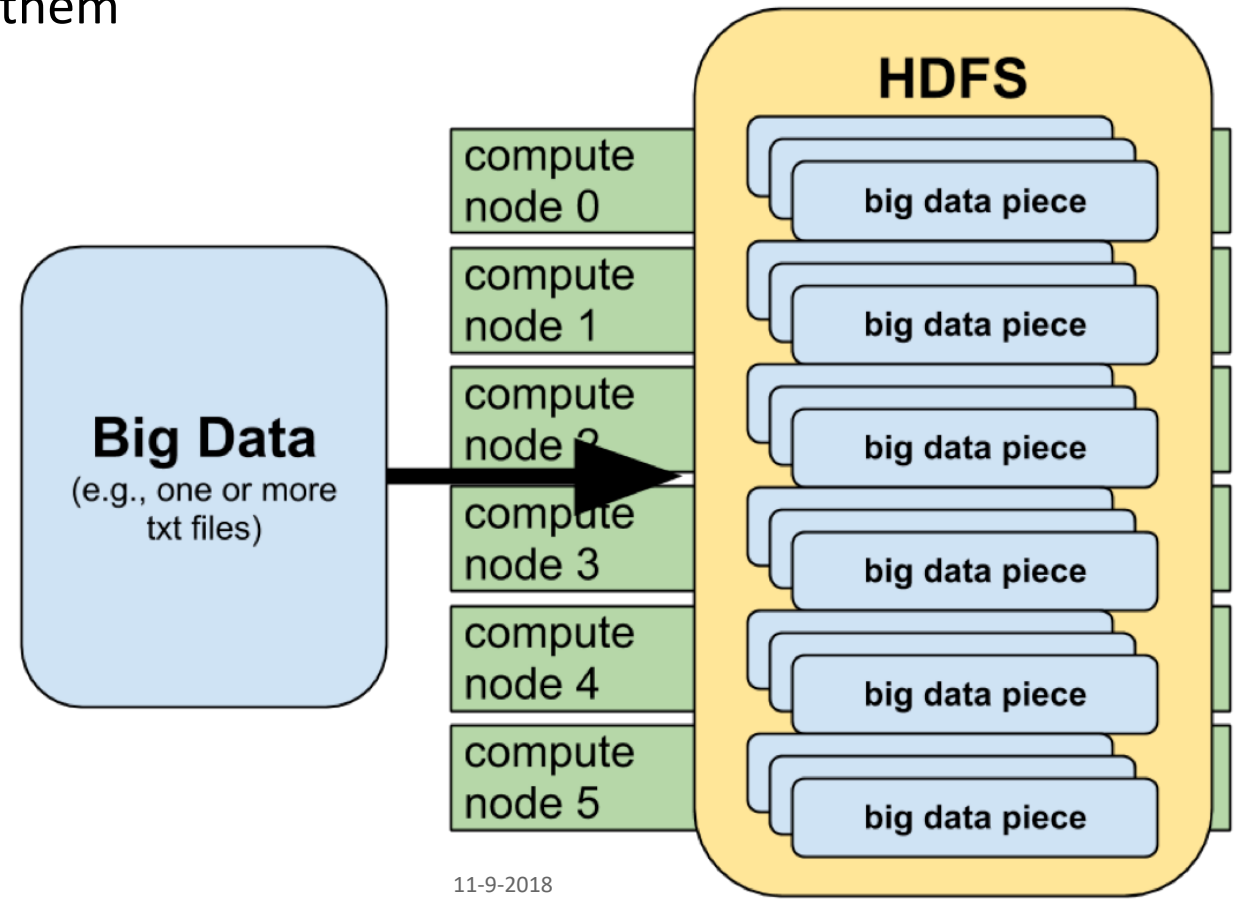
Comparing Map-Reduce to Traditional Parallelism

- Hadoop accomplishes this by storing data in a replicated and distributed fashion on HDFS
  - HDFS stores files in chunks which are physically stored on multiple compute nodes
  - HDFS still presents data to users and applications as single continuous files despite the above fact



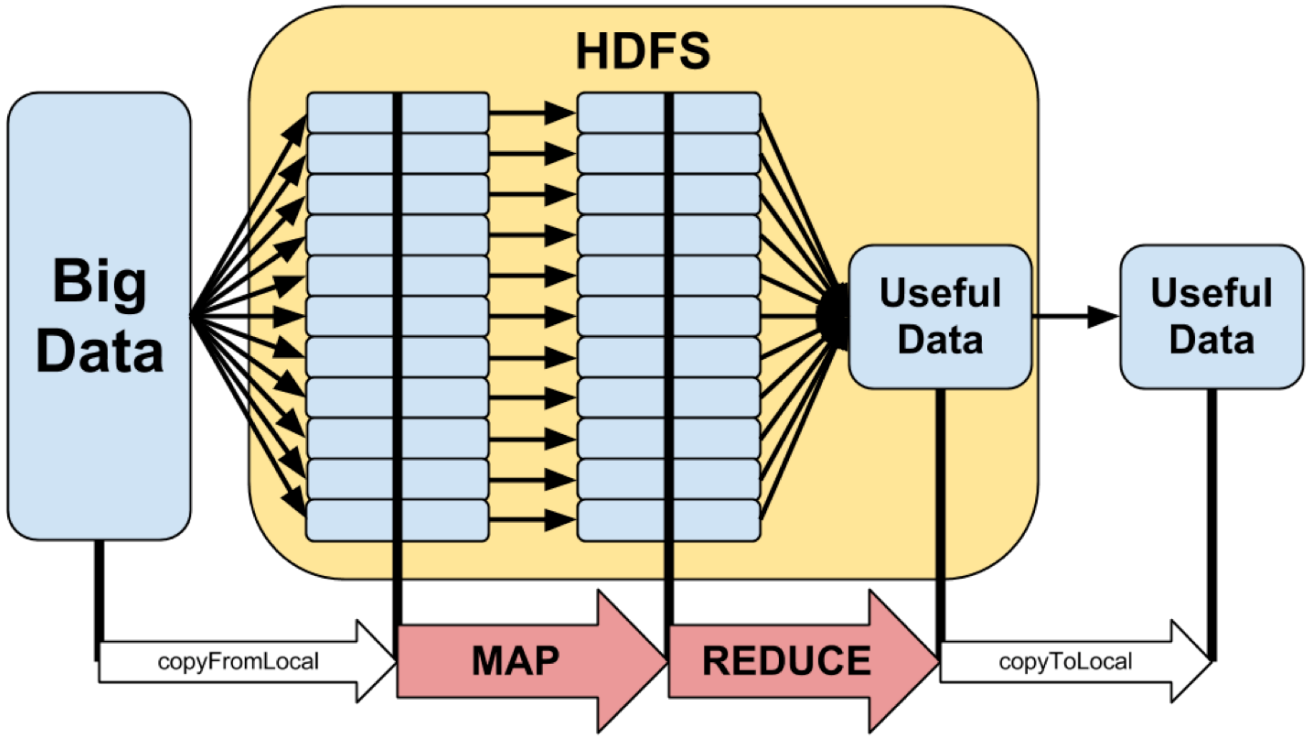
Hadoop - A Map-Reduce Implementation

› Map-reduce is ideal for operating on very large, flat (unstructured) datasets and perform trivially parallel operations on them



Map-reduce jobs

- Hadoop jobs go through a map stage and a reduce stage where
  - the mapper transforms the raw input data into key-value pairs where multiple values for the same key may occur
  - the reducer transforms all of the key-value pairs sharing a common key into a single key with a single value

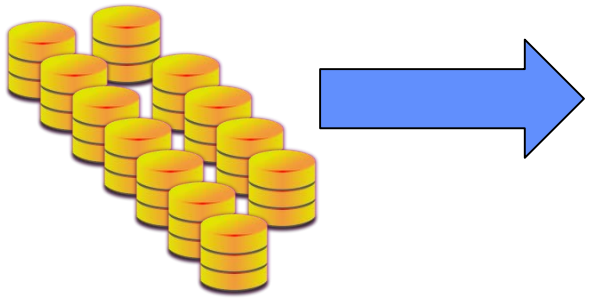


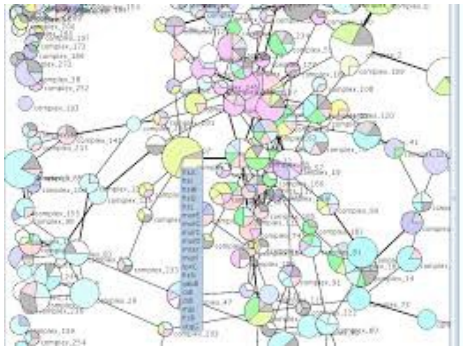
Why Hadoop? USPs

- Moving Computation to Data
  - K*eep* all data: schema-on-read style
- **Scalability**
- **Reliability**
  - Hardware fails
- Key issue?

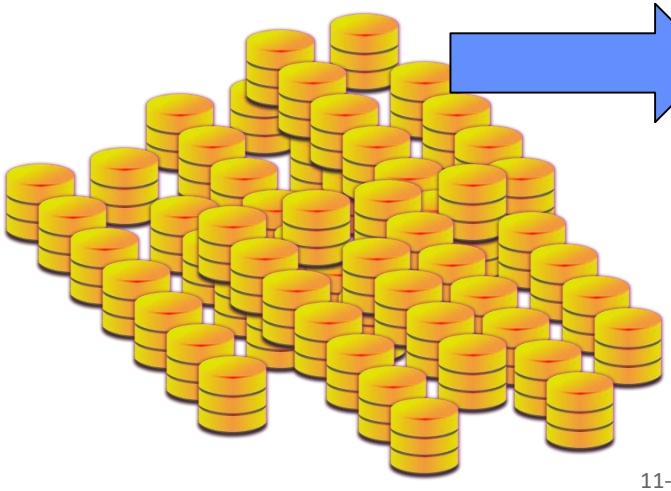
New Kinds of Analysis…

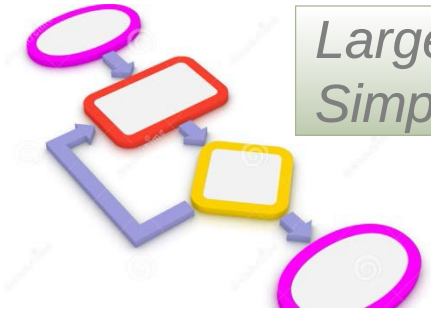
Small Data & Complex Algorithm





Vs.

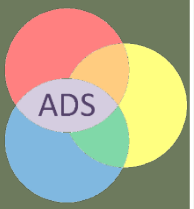




Large Data & Simple Algorithm

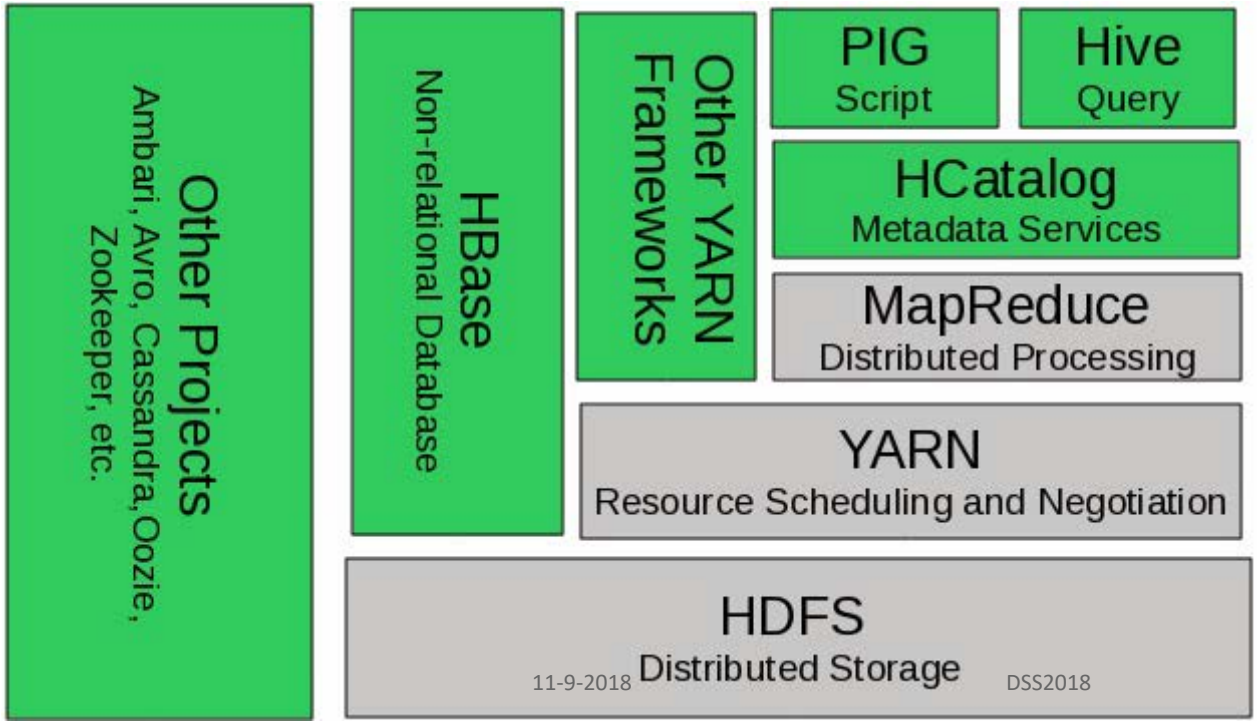
Step by step…

- The Apache framework components



Apache Framework Basic Modules

- Hadoop Common
- Hadoop Distributed File System (HDFS)
- Hadoop YARN
- Hadoop MapReduce



https://hadoop.apache.org/

The project includes these modules:

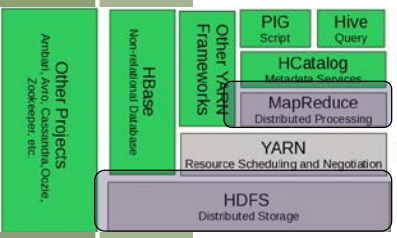
- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop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

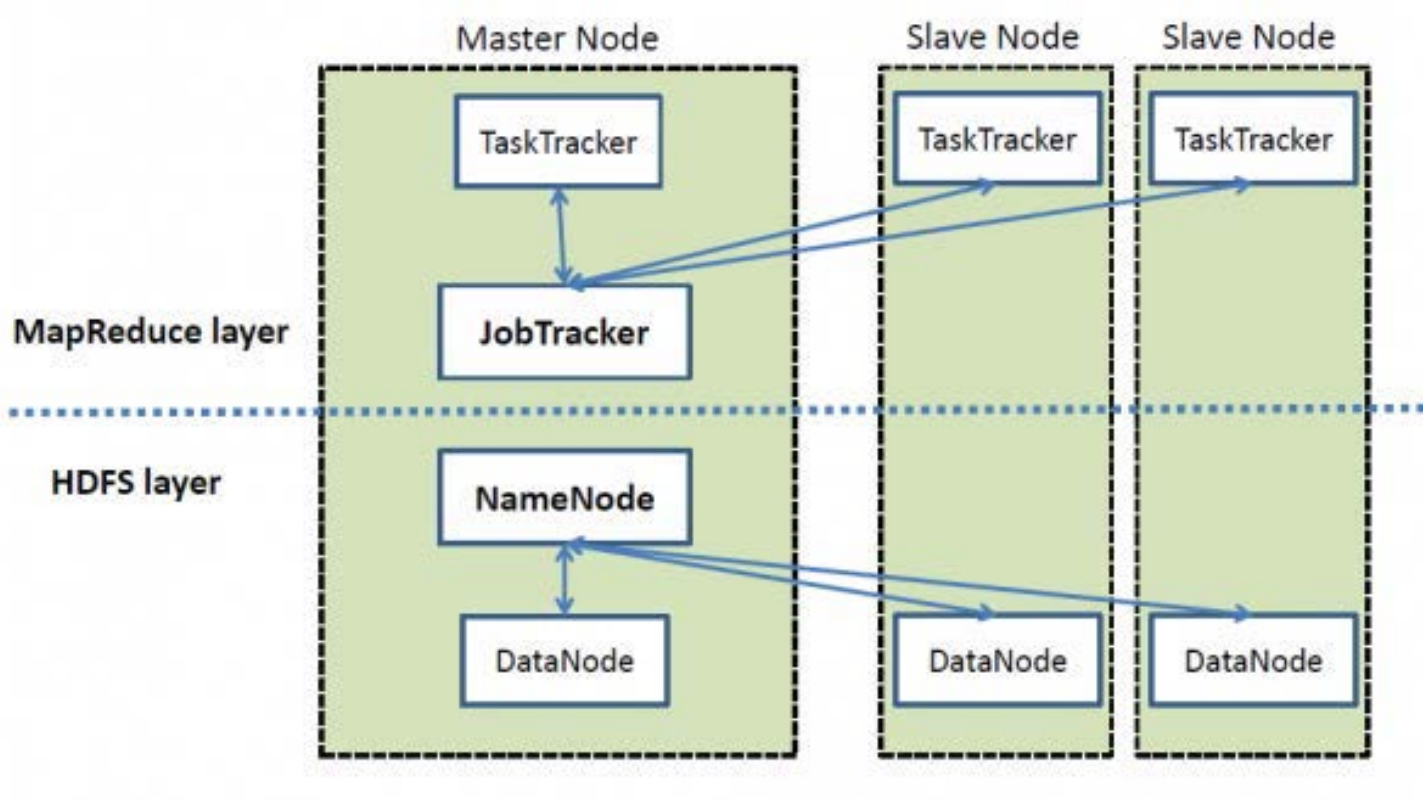
Other Hadoop-related projects at Apache include:

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **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.
- **Mahout™**: A Scalable machine learning and data mining library.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation.
- **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.
- **Tez™**: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases. Tez is being adopted by Hive™, Pig™ and other frameworks in the Hadoop ecosystem, and also by other commercial software (e.g. ETL tools), to replace Hadoop™ MapReduce as the underlying execution engine.
- **ZooKeeper™**: A high-performance coordination service for distributed applications.

Key Terminology within Basic Modules

› Storage & processing

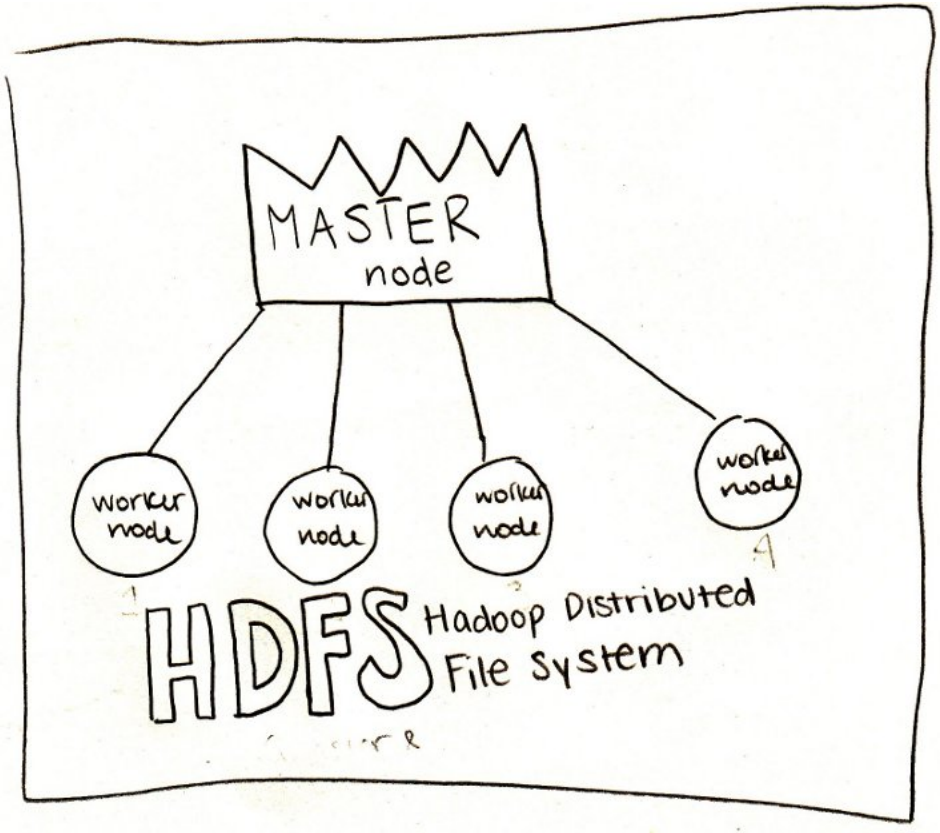




Microsoft Cairo

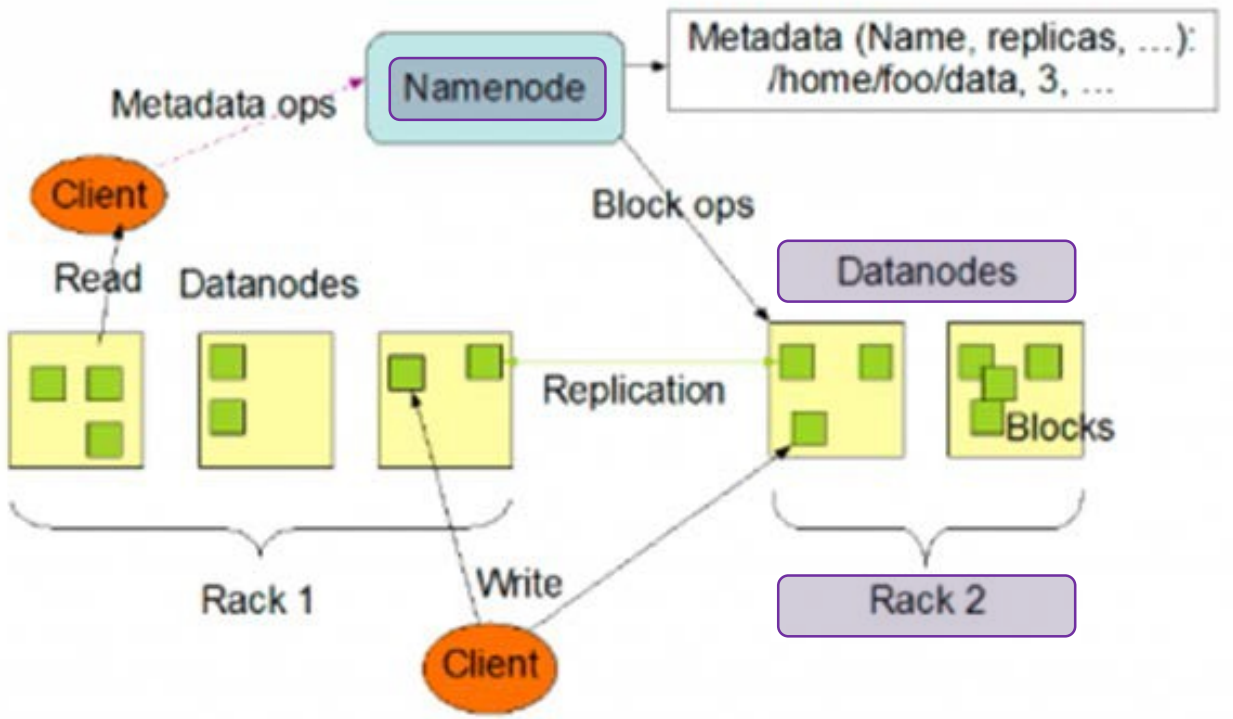
Developer	Microsoft
Working state	Historic, never released
Released to manufacturing	1993, but later cancelled

HDFS



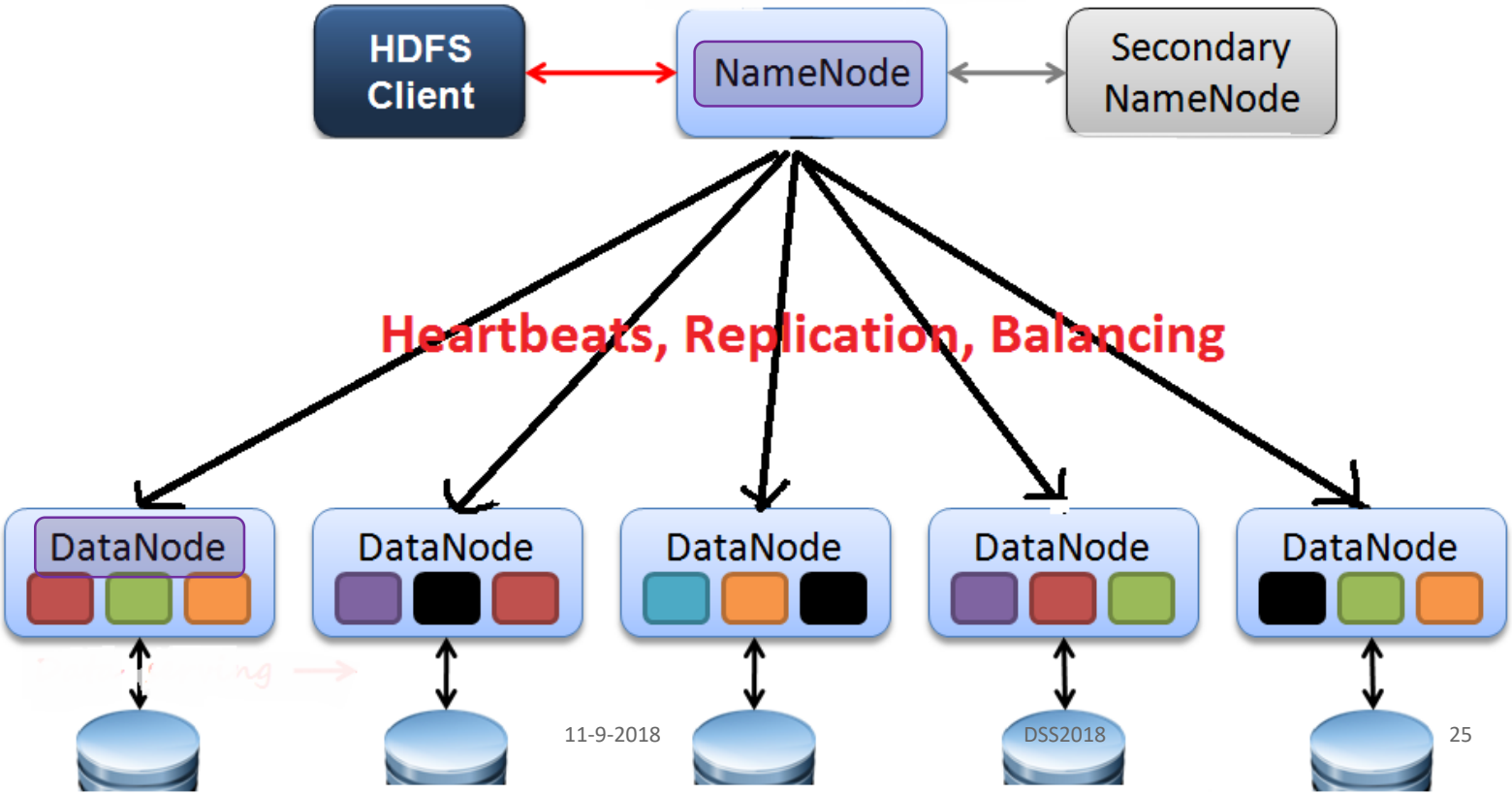
What is HDFS?

› Hadoop Distributed File System: A distributed, scalable, and portable file-system written in Java for the Hadoop framework

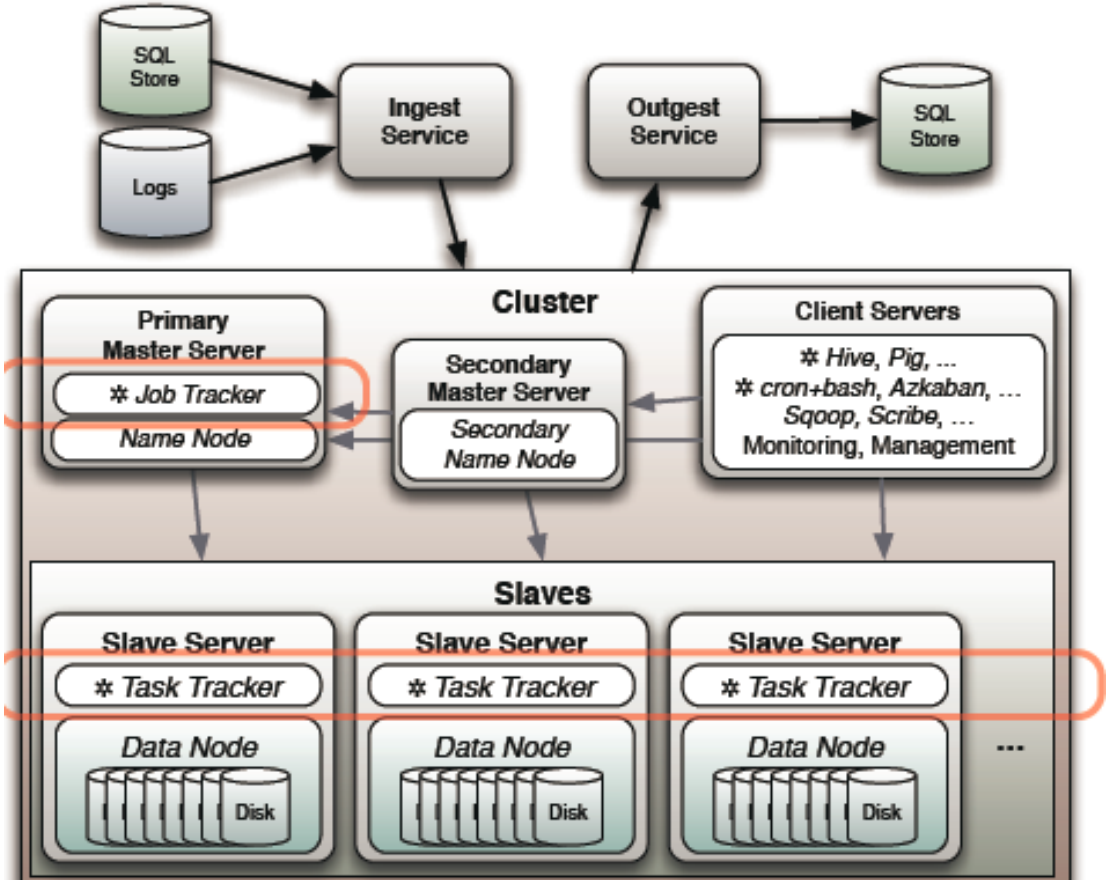


HDFS model

- Fault-tolerant
  - without RAID (Redundant array of independent disks)
  - With commodity hardware



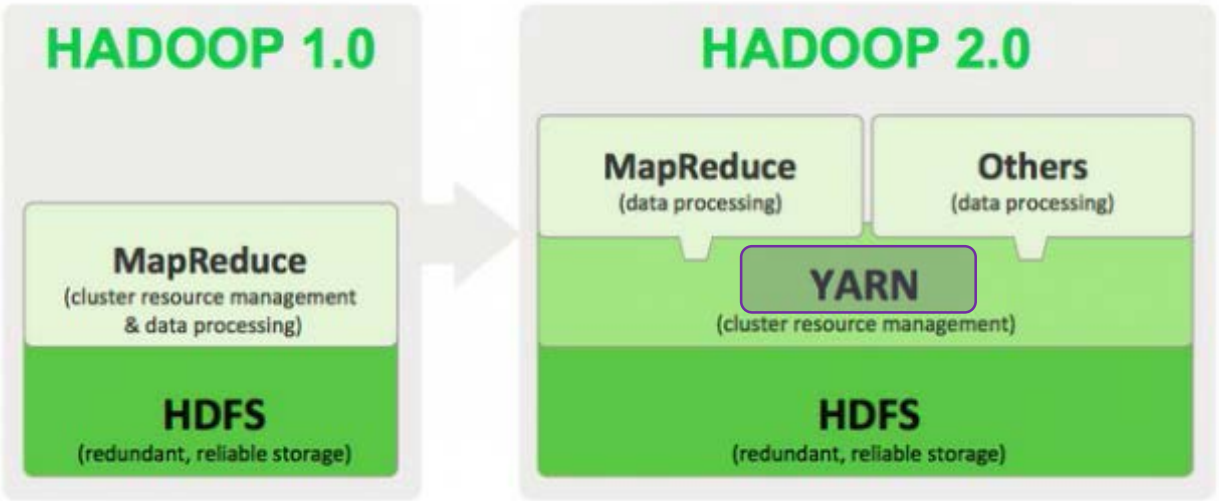
Map-reduce model

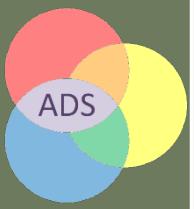


› *Hadoop Streaming:* A utility to enable Map Reduce code in many languages like C, Perl, Python, C++, Bash, etc.
  – e.g. a Python mapper or an AWK reducer.

YARN

Yet Another Resource Negotiator





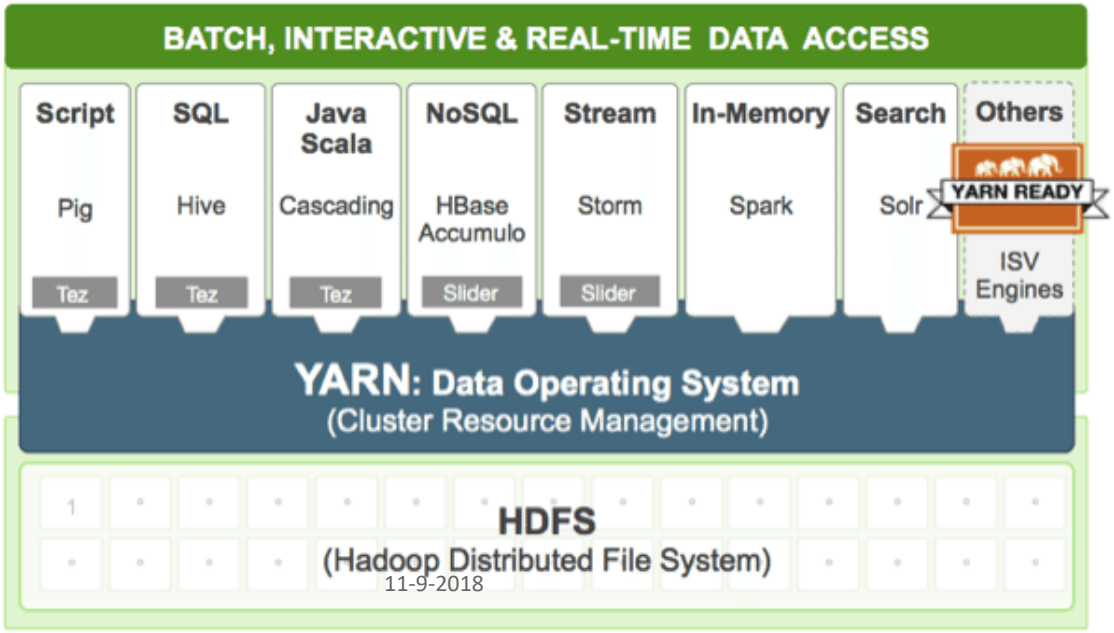
What is Yarn?

› YARN = Apache Hadoop NextGen MapReduce

› YARN enhances the power of a Hadoop compute cluster

– Scalability
  › Multi-tenancy
– Improved cluster utilization
  › Capacity, Guarantees, Fairness, SLAs

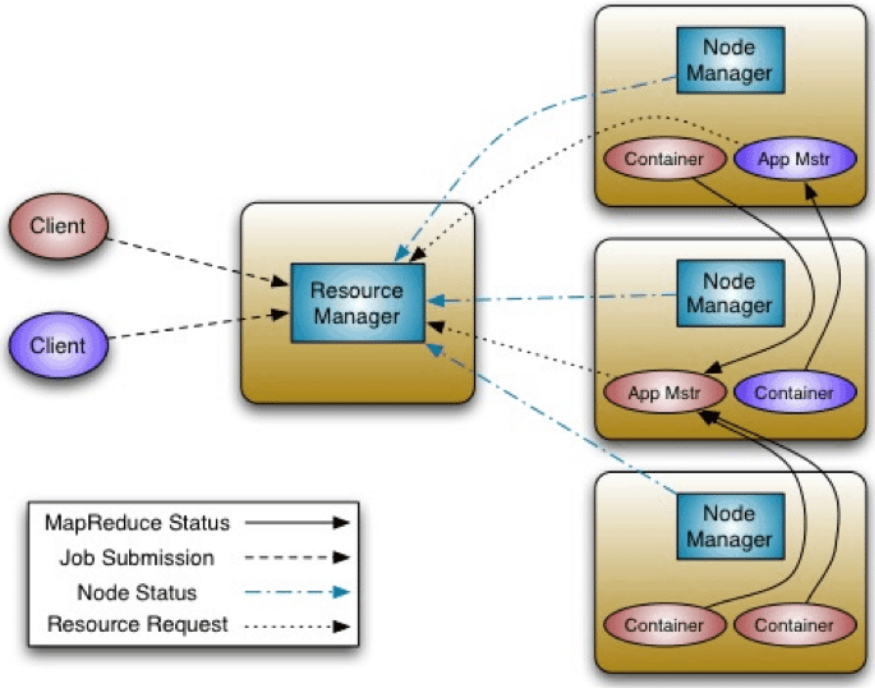
- MapReduce Compatibility
- Supports Other Workloads
  - Graph processing, iterative modelling (ML), ...



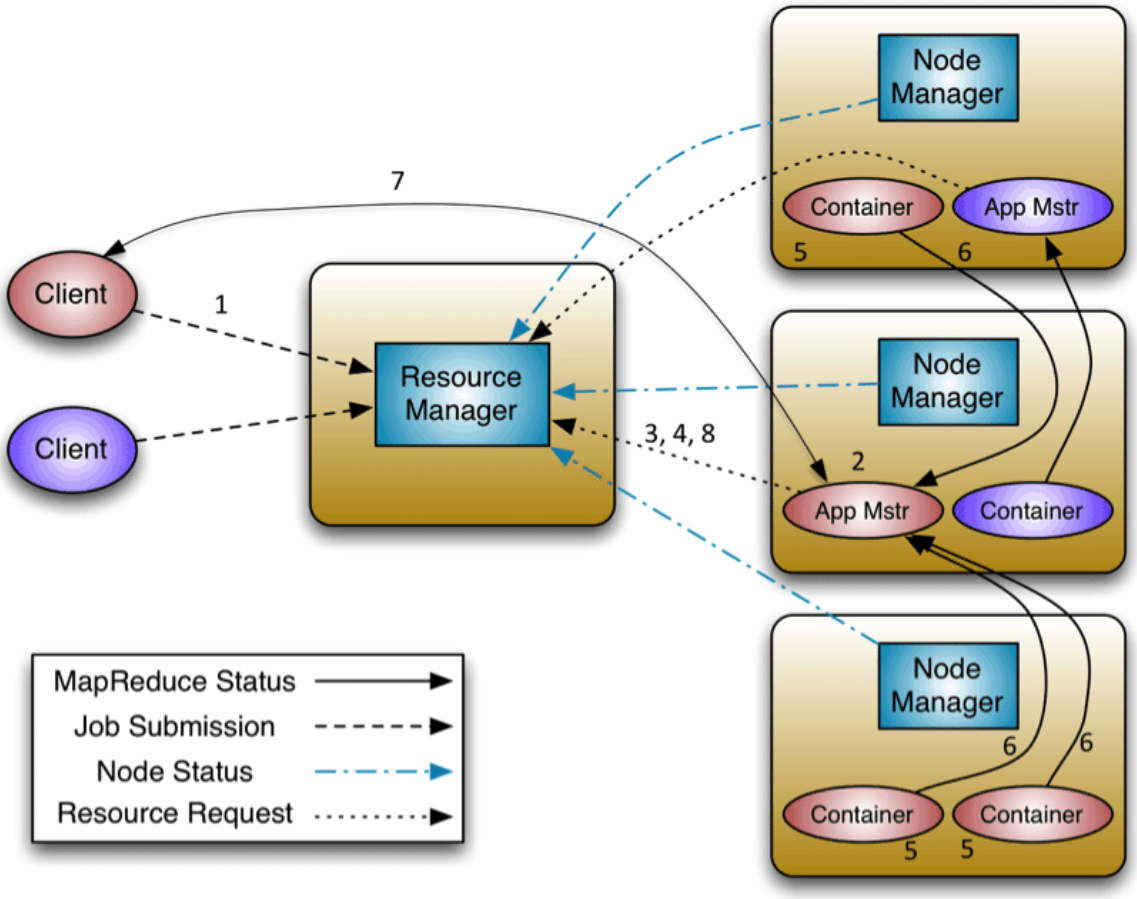
YARN fundamentally

- To split up two major functionalities of the jobtracker,
  - resource management, and
  - job scheduling and monitoring

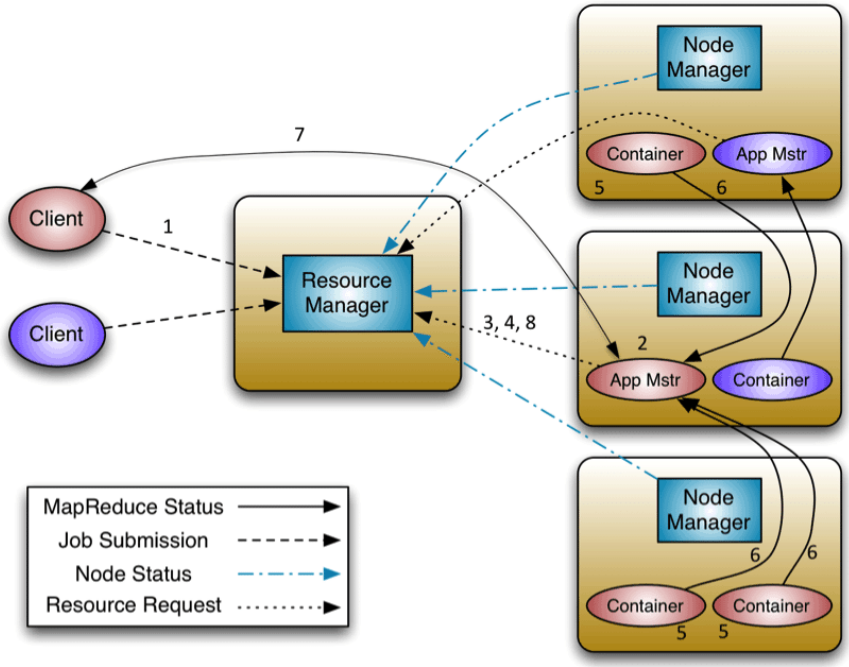
  … into two separate units.
- The idea is to have:
  - one global resource manager, and
  - per application master manager.



YARN

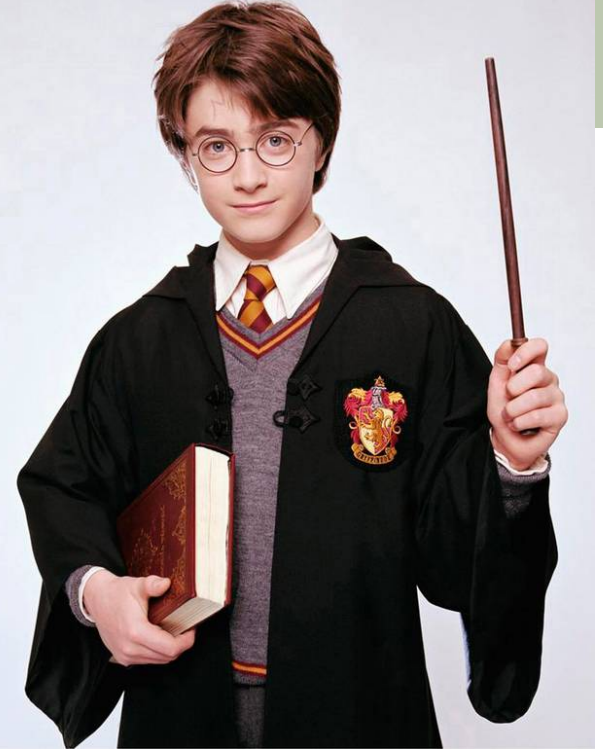

Node Manager
Container
App Mstr
5
6
7
Client
1
Resource Manager
Client
3, 4, 8
2
Node Manager
App Mstr
Container
Node Manager
6
6
Container
Container
5
5
MapReduce Status
Job Submission
Node Status
Resource Request

A YARN application execution sequence



1. A client program *submits* the application, to *launch the application-specific ApplicationMaster* itself.
2. The ResourceManager assumes the responsibility to negotiate a specified container in which to start the ApplicationMaster and then *launches* the ApplicationMaster.
3. The ApplicationMaster, on boot-up, *registers* with the ResourceManager – the registration allows the client program to query the ResourceManager for details, to directly communicate with its own ApplicationMaster.
4. During normal operation the ApplicationMaster negotiates appropriate resource containers.
5. On successful container allocations, the ApplicationMaster launches the container by providing the container launch specification to the NodeManager. The launch specification, typically, includes the necessary information to allow the container to communicate with the ApplicationMaster itself.
6. The application code executing within the container then provides necessary information (progress, status etc.) to its ApplicationMaster via an *application-specific protocol*.
7. During the application execution, the client that submitted the program communicates directly with the ApplicationMaster to get status, progress updates etc. via an application-specific protocol.
8. Once the application is complete, and all necessary work has been finished, the ApplicationMaster deregisters with the ResourceManager and shuts down, allowing its own container to be repurposed.

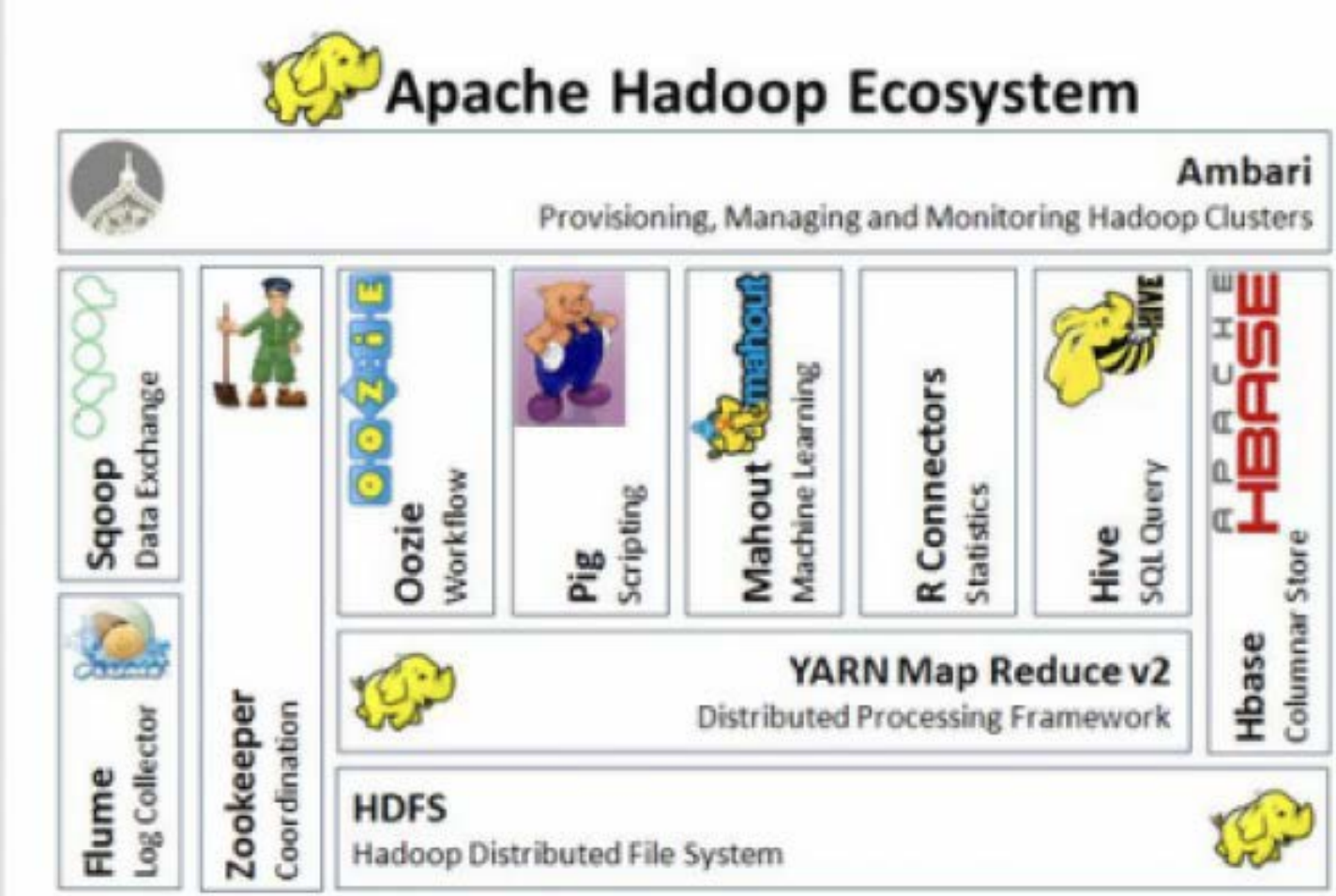
https://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/



The Hadoop Zoo

A Big Data Technologies Ecosystem

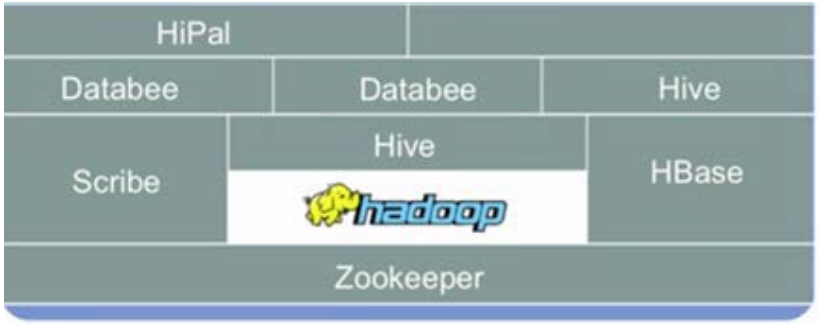
The Hadoop Zoo



Hadoop for Big Data in business

› Original Google Stack

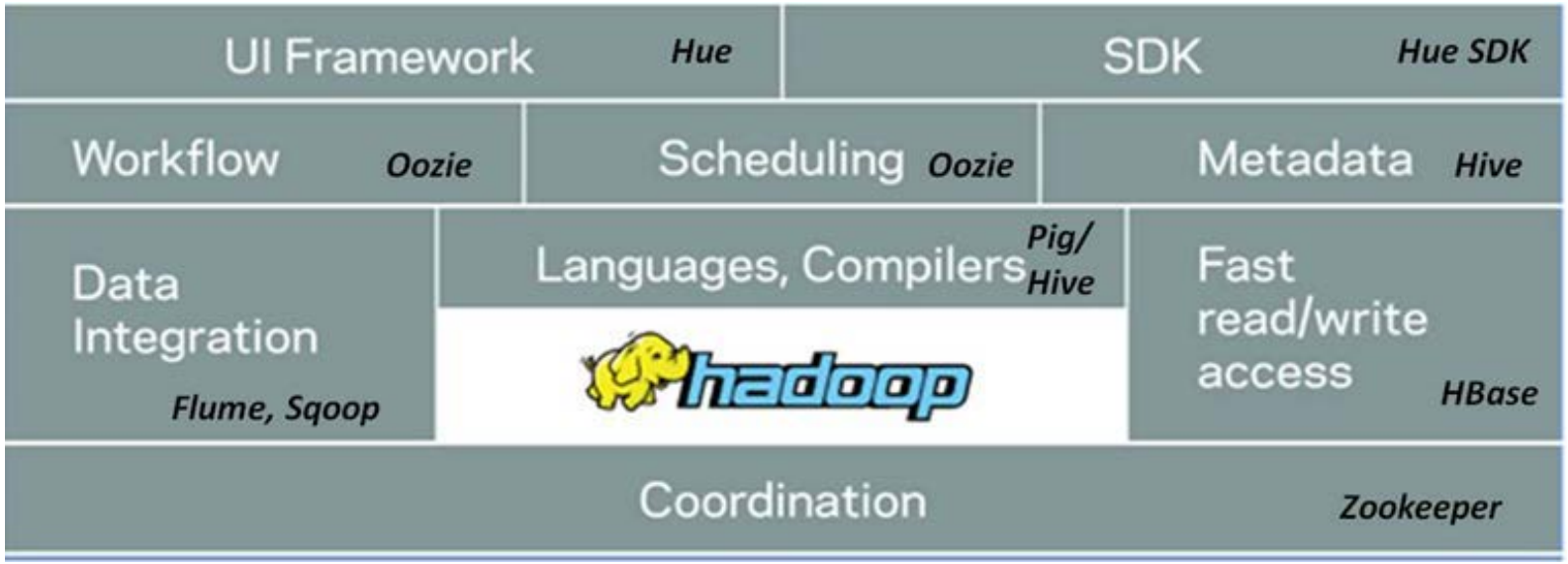
› Facebook Stack



› Yahoo Stack

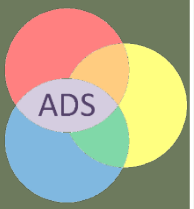
› LinkedIn Stack

Cloudera's Hadoop stack



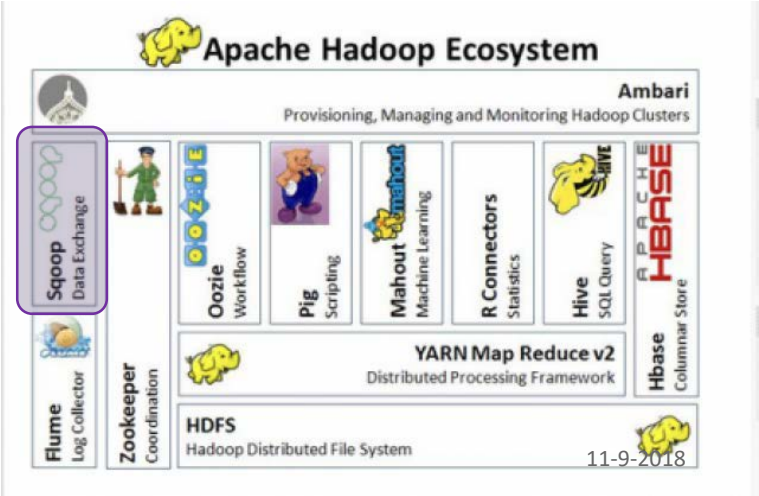
The Hadoop Ecosystem

Major components on top of the Hadoop framework in Cloudera's stack



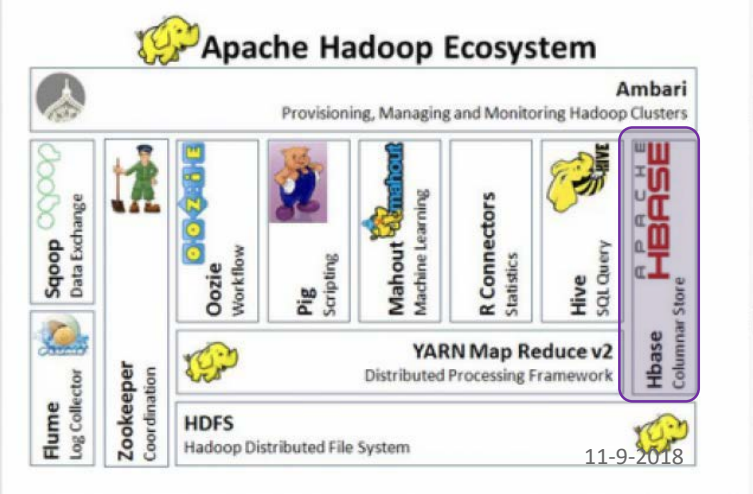
Apache Sqoop

- SQL to Hadoop
- Tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases



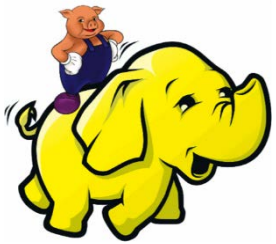
Apache HBase

- Column-oriented database management system
- Key-value store
- Based on Google Big Table
- Can hold extremely large data
- Dynamic data model
- Not a Relational DBMS



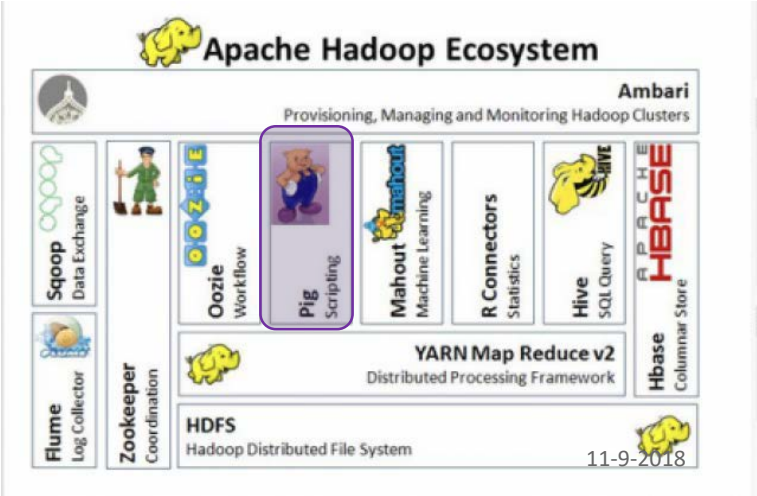
Chang, et al. (2006). Bigtable: A Distributed Storage System for Structured Data

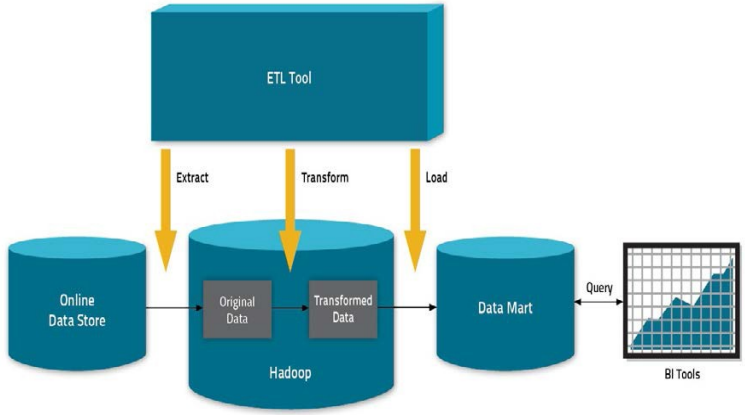
"A Bigtable is a sparse, distributed, persistent multi-dimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an uninterpreted array of bytes."



Apache Pig

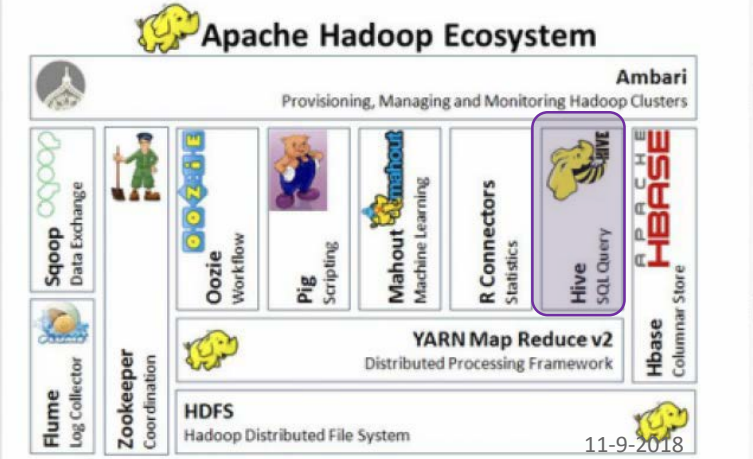
- High level programming on top of Hadoop MapReduce
- The language: Pig Latin
- Data analysis problems as data flows
- Originally developed at Yahoo 2006





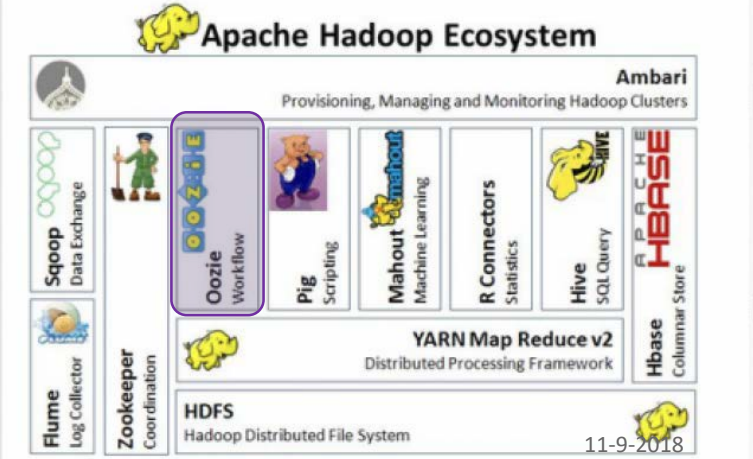
Apache Hive

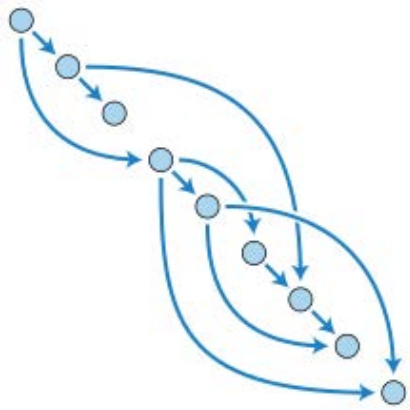
- Data warehouse software facilitates querying and managing large datasets residing in distributed storage
- SQL-like language
- Facilitates querying and managing large datasets in HDFS
- Mechanism to project structure onto this data and query the data using a SQL-like language called **HiveQL**



Apache Oozie

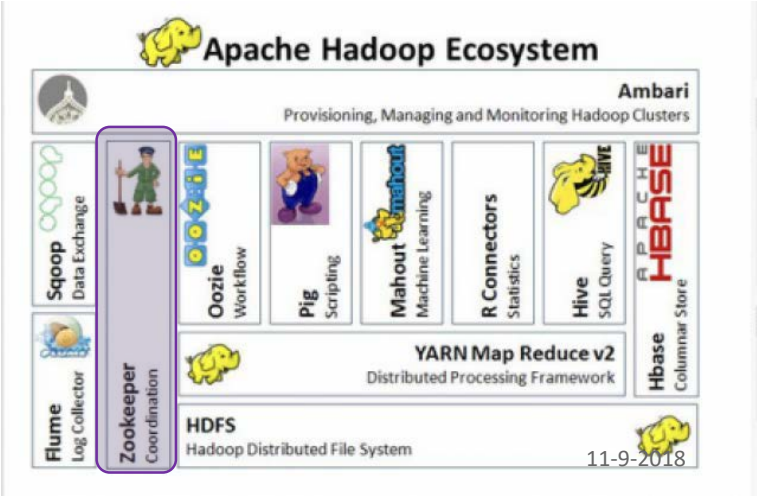
- Workflow scheduler system to manage Apache Hadoop jobs
  - doesn't replace your scheduler but adds if-then-else branching and control with Hadoop jobs
- Oozie workflow jobs are DAGs or Directed Graphs
- Oozie coordinator jobs are recurrent Oozie workflow jobs that are triggered by frequency or data availability
- Supports MapReduce, Pig, Apache Hive, and Sqoop, etc.





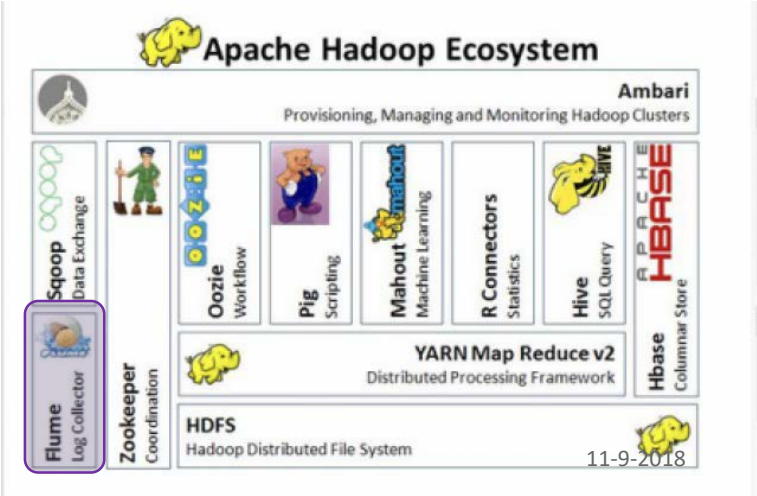
Apache ZooKeeper

- Provides operational services for a Hadoop cluster group services
- Centralized service for:
  - maintaining configuration information naming services
  - providing distributed synchronization and providing group services

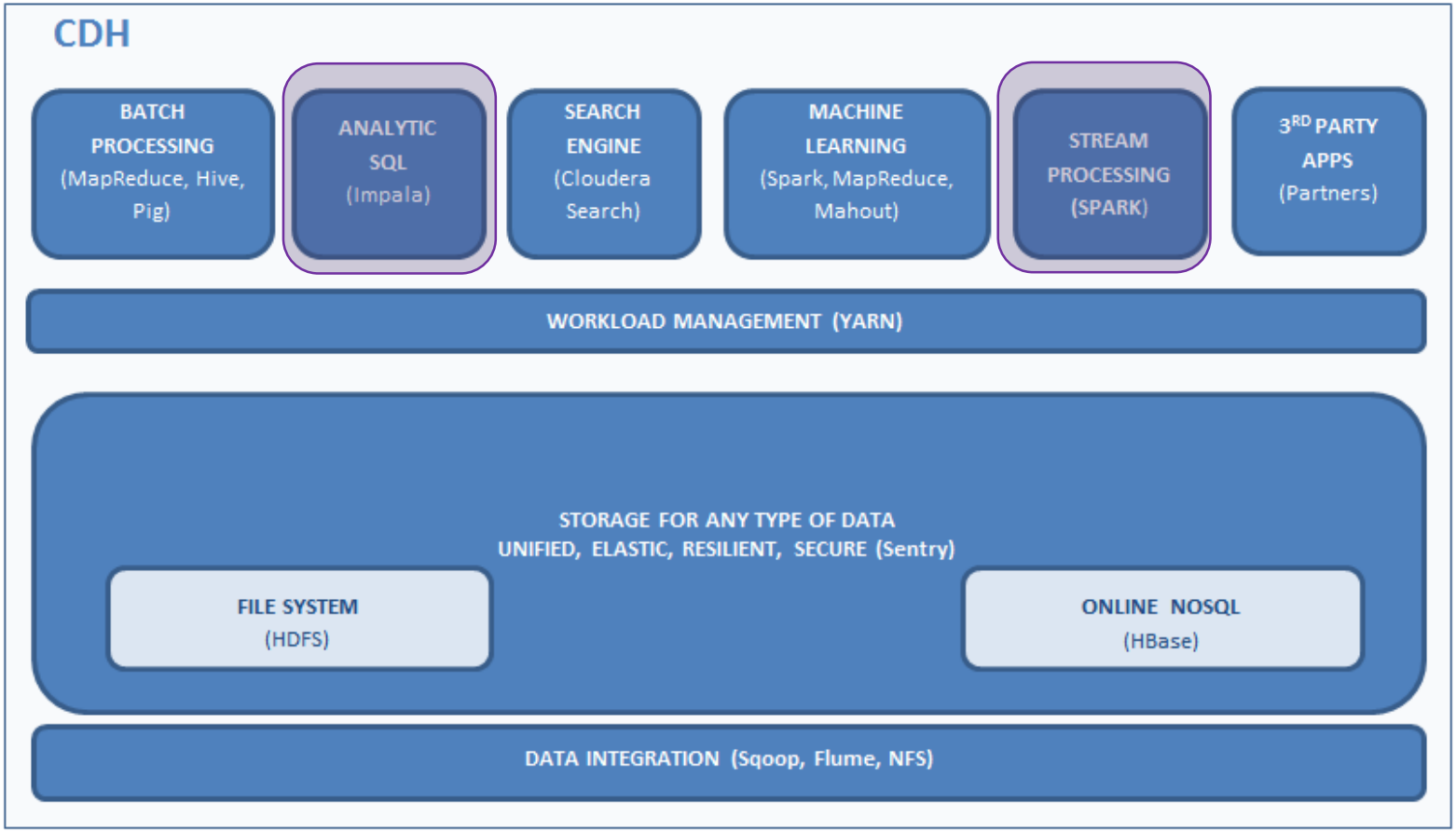


Apache Flume

- Distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data
- A real time loader for streaming your data into Hadoop.
- Possible sources for Flume include Avro, files, and system logs
- Possible sinks include HDFS and HBase.

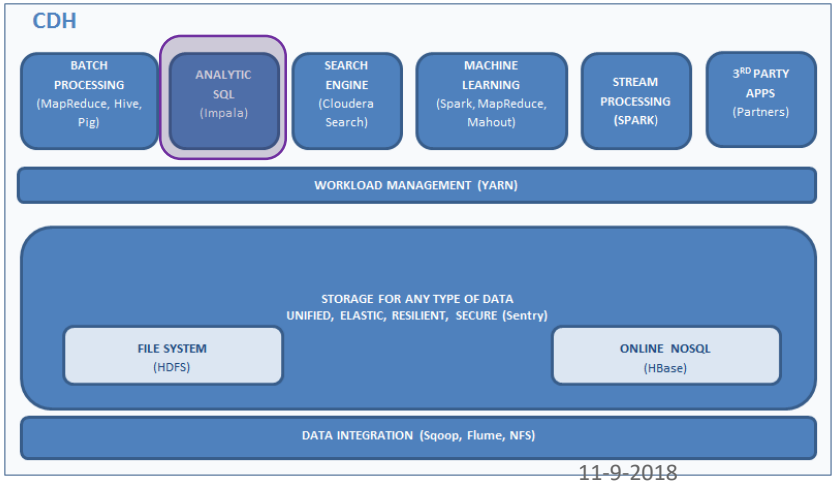


Additional Cloudera Data Hub components



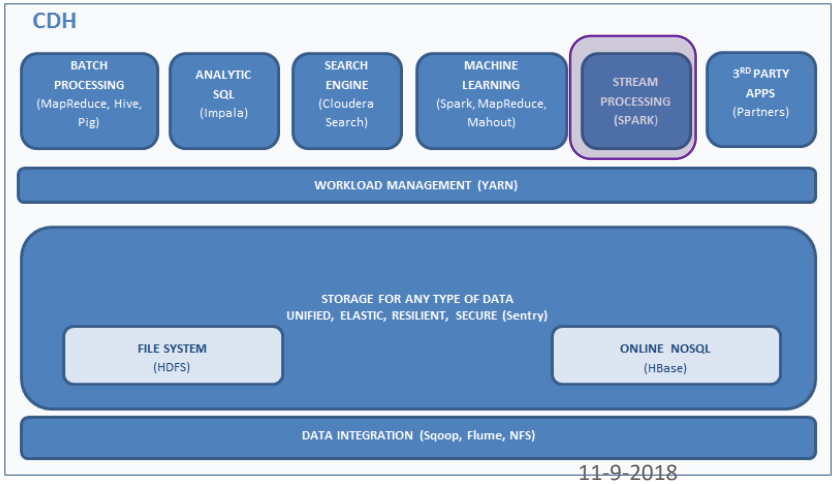
Cloudera Impala

- Open source massively parallel processing (MPP) SQL query engine for Apache Hadoop
- Inspired by Google's paper on Dremel
- Facilitates real-time querying of data in HDFS or HBase
  - Uses an SQL-like language
  - The secret behind Impala's speed: it "circumvents Map Reduce to directly access the data through a specialized distributed query engine very similar to commercial parallel RDBMSs."



Apache Spark

- Multi-stage in-memory primitives provides performance up to 100 times faster for certain applications
- Allows user programs to load data into a cluster's memory and query it repeatedly
  - Well-suited for machine learning



This week's Literature readings

- White, J. (2016). *Hadoop: The Definitive Guide*. Third edition. O'Reilly.
  - CH 1: Meet Hadoop
  - …

  - *Please remember last week's Literature readings ;-)*