Introduction to Hadoop

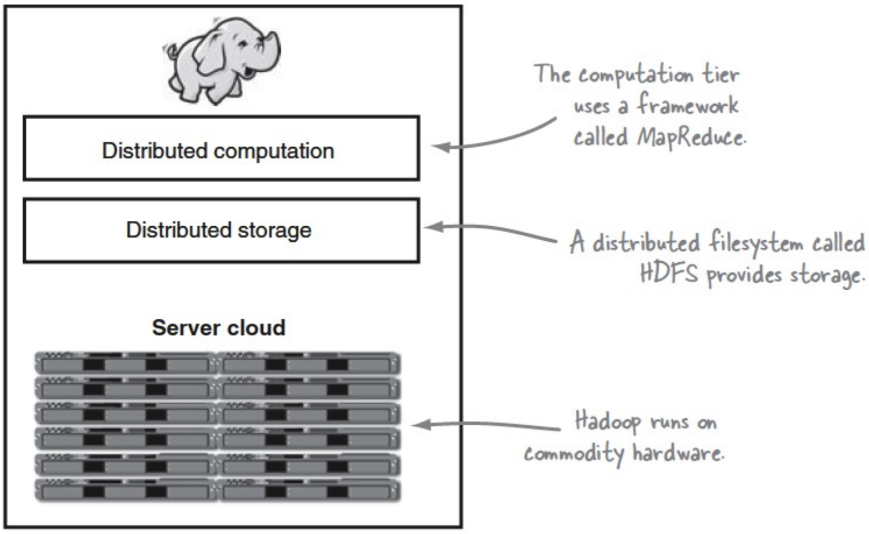
Assoc. Prof. Dr. Thanachart Numnonda
Executive Director
IMC Institute
April 2016

What is Hadoop?

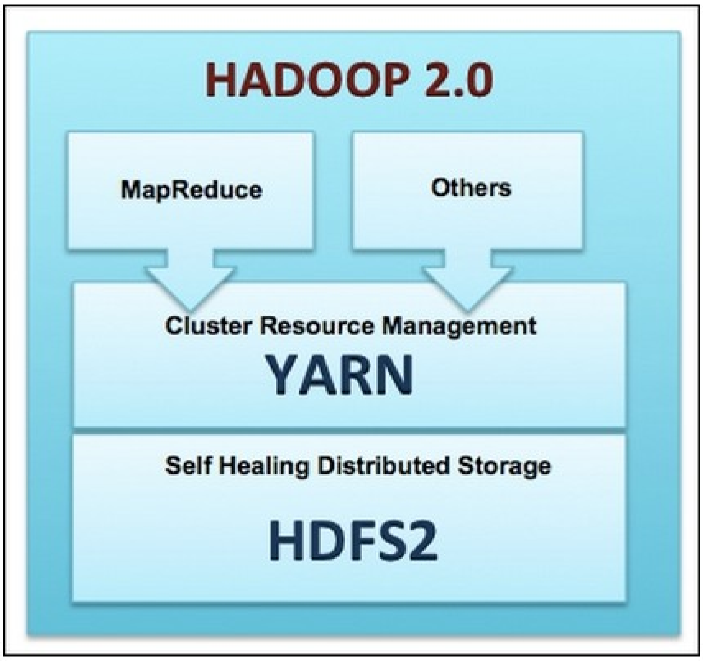
A scalable fault-tolerant distributed system for data storage and processing

Completely written in java
Open source & distributed under Apache license

Hadoop Environment

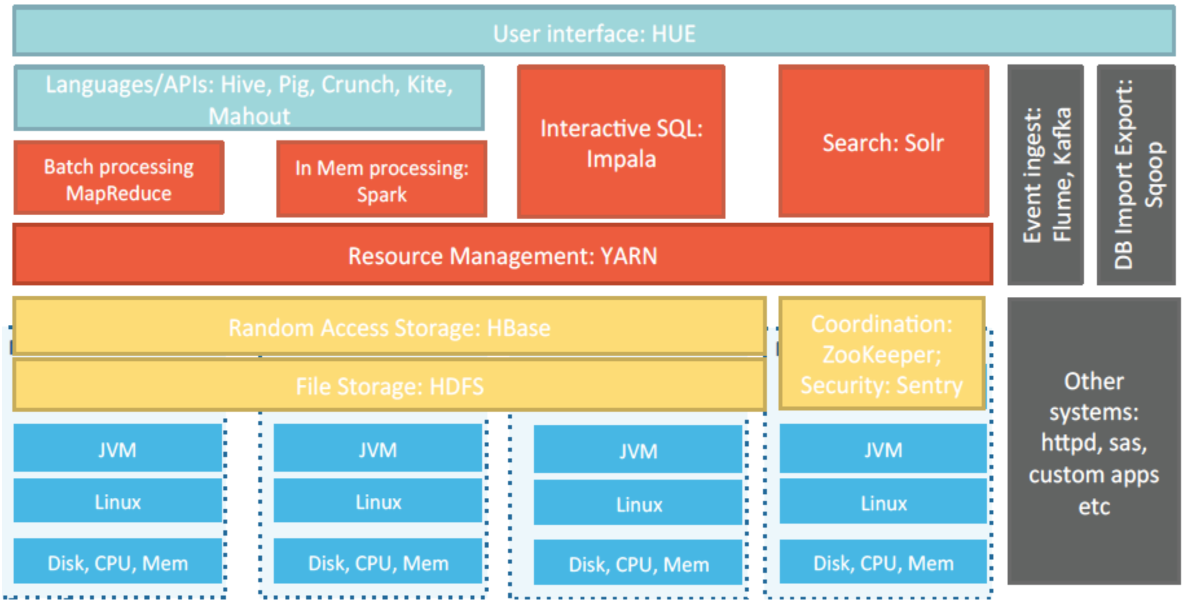

Distributed computation
Distributed storage
Server cloud
The computation tier uses a framework called MapReduce.
A distributed filesystem called HDFS provides storage.
Hadoop runs on commodity hardware.

Source: Hadoop in Practice; Alex Holmes



Source: HDInsight Essentials - Second Edition

Hadoop Ecosystem



User interface: HUE

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

Coordination: ZooKeeper; Security: Sentry

JVM | Linux | Disk, CPU, Mem

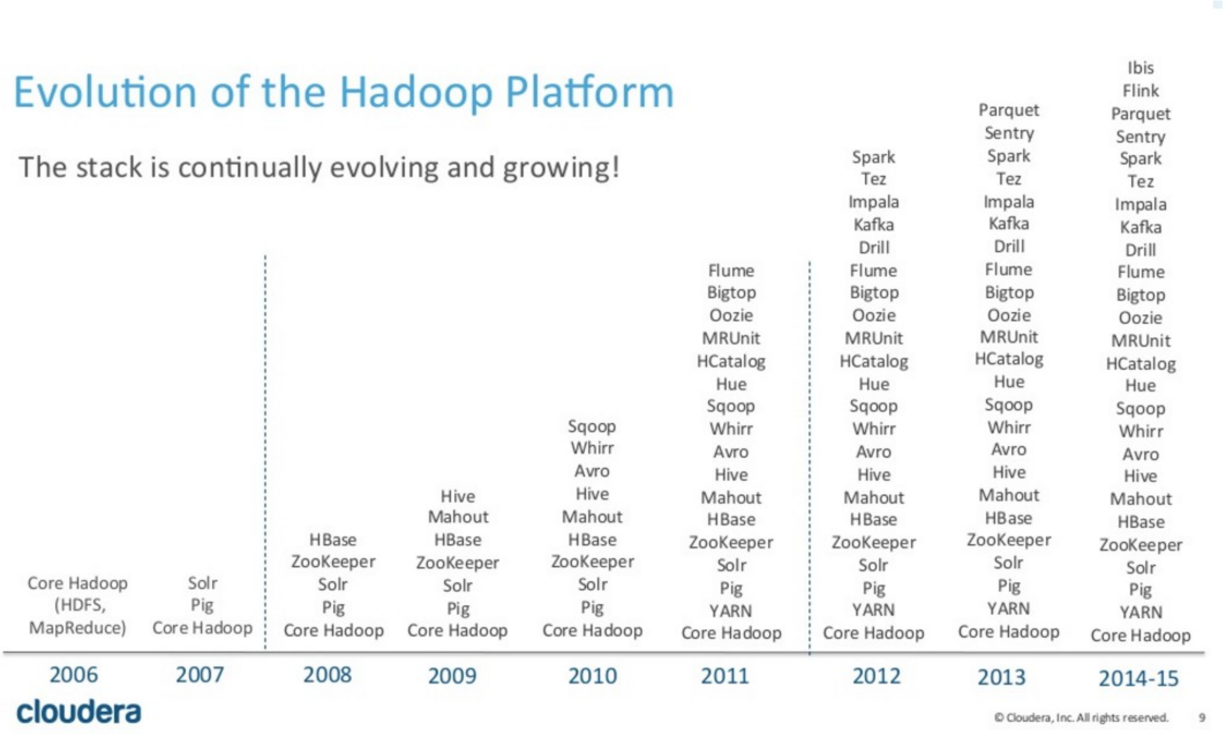
JVM | Linux | Disk, CPU, Mem

JVM | Linux | Disk, CPU, Mem

JVM | Linux | Disk, CPU, Mem

Other systems: httpd, sas, custom apps etc

Source: Apache Hadoop Operations for Production Systems, Cloudera



Evolution of the Hadoop Platform

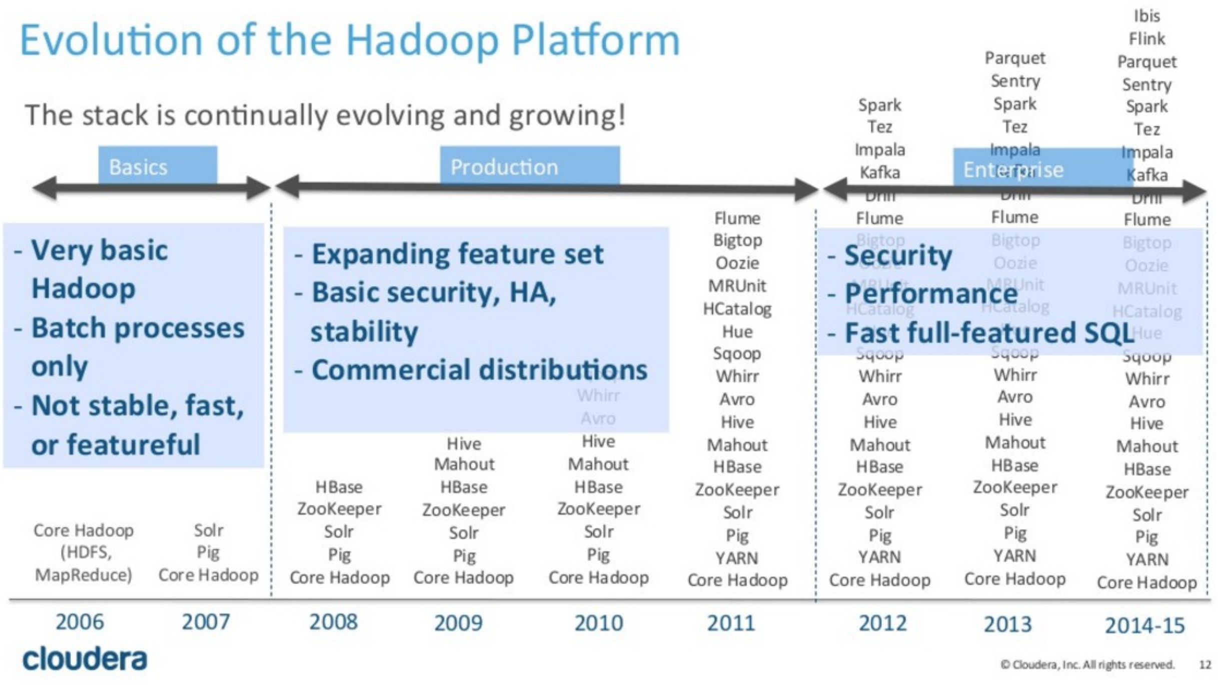
The stack is continually evolving and growing!

2006	2007	2008	2009	2010	2011	2012	2013	2014-15
Core Hadoop (HDFS, MapReduce)	Solr Pig Core Hadoop	HBase ZooKeeper Solr Pig Core Hadoop	Hive Mahout HBase ZooKeeper Solr Pig Core Hadoop	Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig Core Hadoop	Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop	Spark Tez Impala Kafka Drill Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop	Parquet Sentry Spark Tez Impala Kafka Drill Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop	Ibis Flink Parquet Sentry Spark Tez Impala Kafka Drill Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop

cloudera

© Cloudera, Inc. All rights reserved. 9

Source: The evolution and future of Hadoop storage: Cloudera



Evolution of the Hadoop Platform

The stack is continually evolving and growing!

Basics

- Very basic Hadoop
- Batch processes only
- Not stable, fast, or featureful

Production

- Expanding feature set
- Basic security, HA, stability
- Commercial distributions

Enterprise

- Security
- Performance
- Fast full-featured SQL

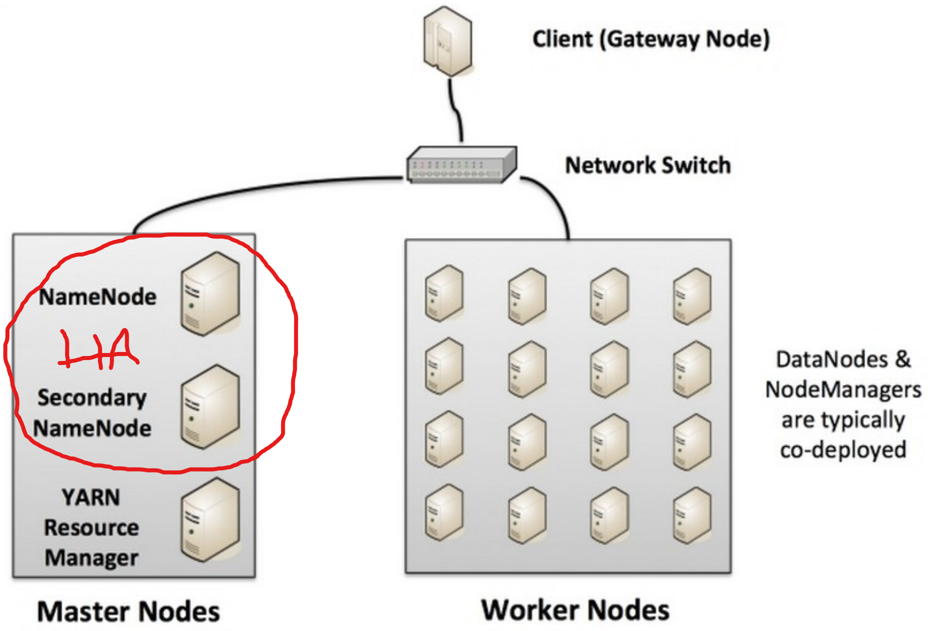
2006	2007	2008	2009	2010	2011	2012	2013	2014-15
Core Hadoop (HDFS, MapReduce)	Solr Pig Core Hadoop	HBase ZooKeeper Solr Pig Core Hadoop	Hive Mahout HBase ZooKeeper Solr Pig Core Hadoop	Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig Core Hadoop	Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop	Spark Tez Impala Kafka Drill Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop	Parquet Sentry Spark Tez Impala Kafka Drill Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop	Ibis Flink Parquet Sentry Spark Tez Impala Kafka Drill Flume Bigtop Oozie MRUnit HCatalog Hue Sqoop Whirr Avro Hive Mahout HBase ZooKeeper Solr Pig YARN Core Hadoop

cloudera

12

Source: The evolution and future of Hadoop storage: Cloudera

Hadoop Cluster



Source: HDInsight Essentials - Second Edition

Data Processing Technology using Hadoop

- MapReduce
- Hive
- Pig
- Imapla
- Drill
- Spark

What is MapReduce?

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster

A MapReduce program is composed of a Map() method and a Reduce() method. It runs on a batch mode.

What is Hive?

Hive is developed by Facebook,
designed to enable easy data summarization,
ad-hoc querying and analysis of large volumes of data.

It provides a simple query language called Hive QL, → Relation DB
which is based on SQL.
It is not designed for online transaction processing
and does not offer real-time queries
and row level updates

What is Pig?

Pig is a platform for analyzing large data sets
that consists of a high-level language
for expressing data analysis programs,
coupled with infrastructure for evaluating these programs.

It is developed by Yahoo and uses a scripting language
Called Pig Latin.

What is Impala?

Impala is a query engine that runs on Apache Hadoop.
It brings scalable parallel database technology
to Hadoop, enabling users to issue
low-latency SQL queries to data stored in HDFS
and Apache HBase without requiring
data movement or transformation.

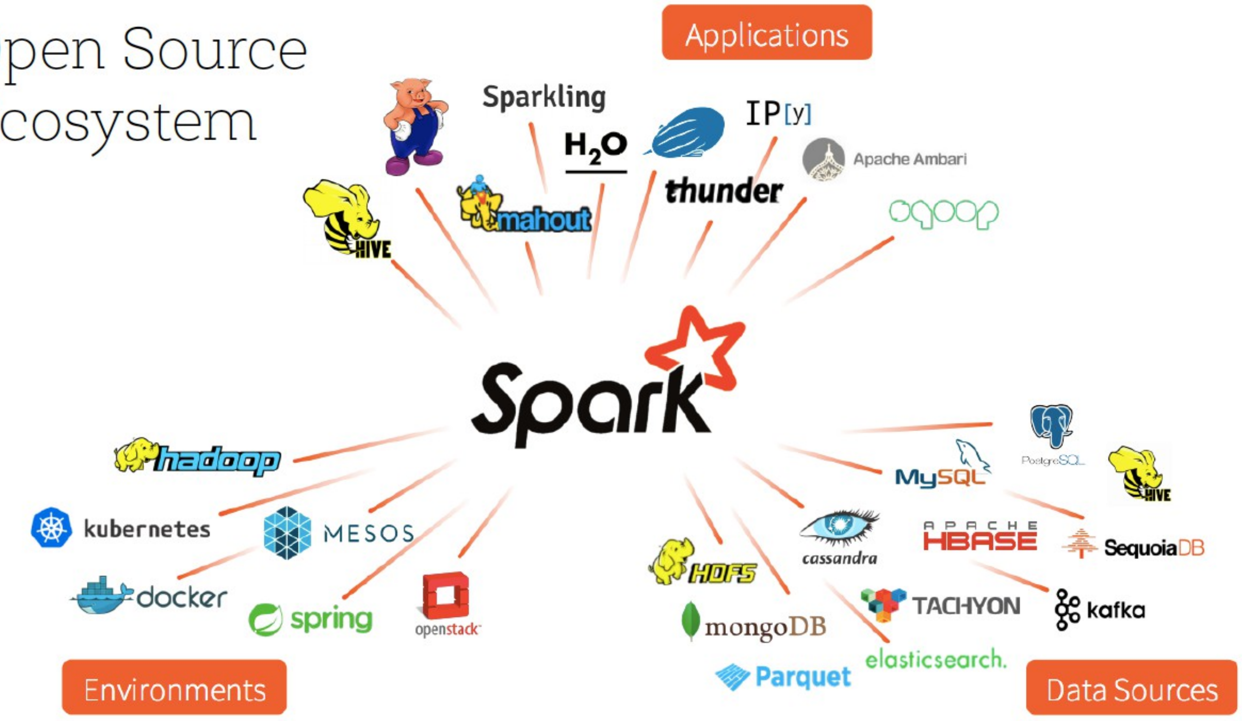
What is Spark?

An open source big data processing framework
built around speed, ease of use,
and sophisticated analytics.

Spark enables applications in Hadoop clusters
to run up to 100 times faster in memory
and 10 times faster even when running on disk.

Spark

Open Source Ecosystem



Applications

Environments

Data Sources

Source: Jump start into Apache Spark and Databricks

Processing Techniques

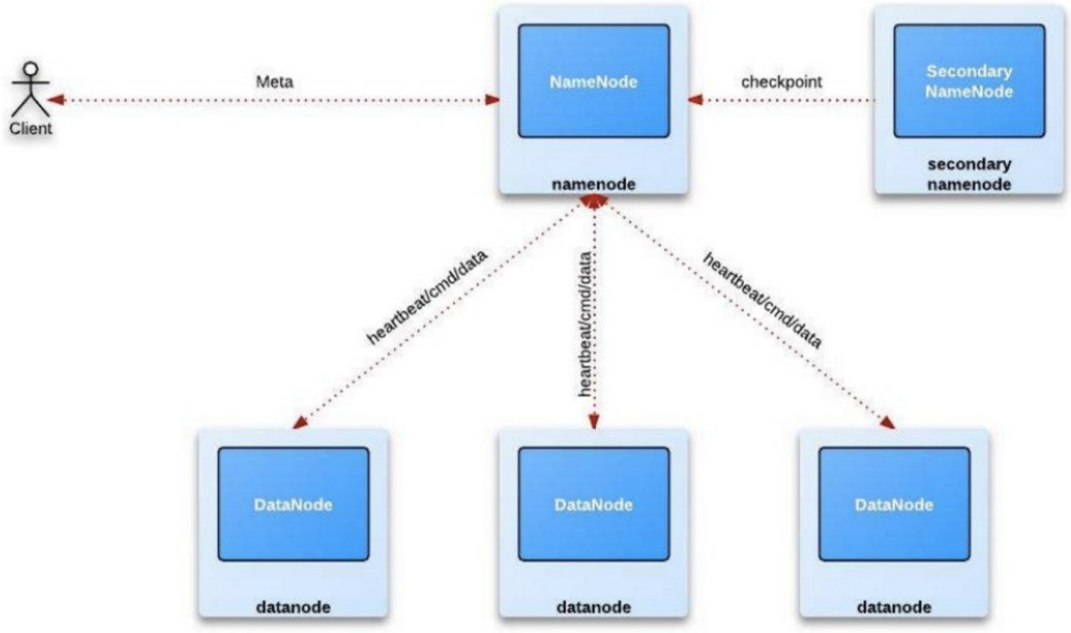
	Hive	Pig	Spark	Impala
Query Latency	Medium (Tez), High	High	Low	Low
Storage	HDFS	HDFS	HDFS,Object Storage	HDFS
Language	Hive QL	Pig Latin	Java, Python, Scala, R	Impala
Analysis	Batch	Batch	Interactive, Real-time (Spark Streaming), Machine Learning (MLlib)	Interactive
SQL Compatibility	Medium	No	Low (Spark SQL)	Medium

Lisp
Haskell

What is HDFS?

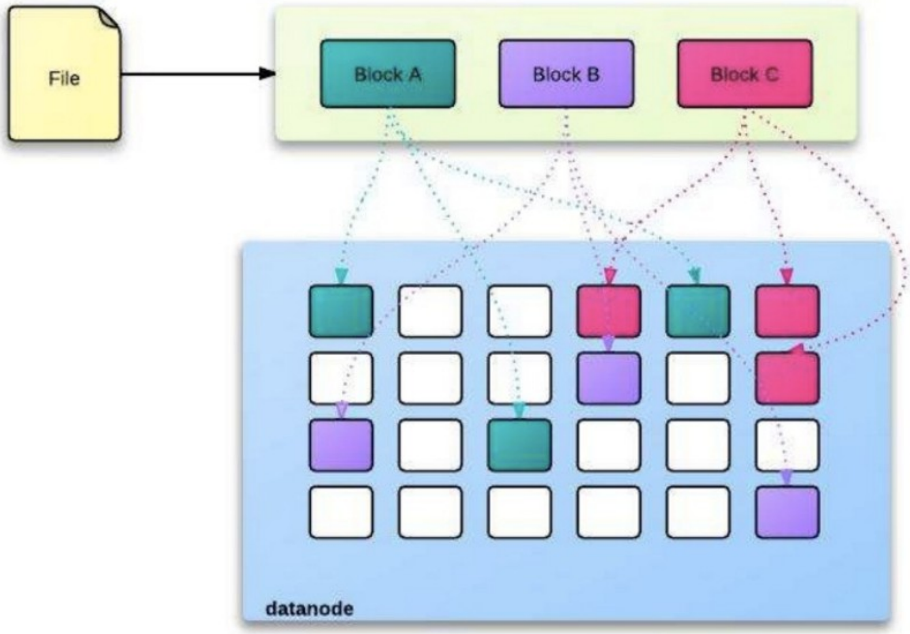
A distributed file system that provides high-performance access to data across Hadoop clusters.
Like other Hadoop-related technologies,
HDFS has become a key tool
for managing pools of big data
and supporting big data analytics applications.

HDFS Architecture



Source: Details intro to HDFS, Clay Jiang

HDFS Storage



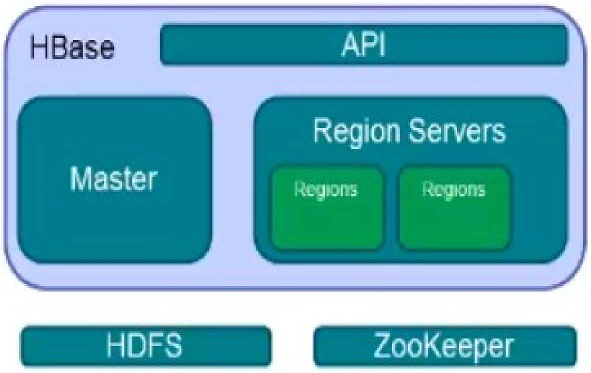
Source: Details intro to HDFS, Clay Jiang

What is HBase?

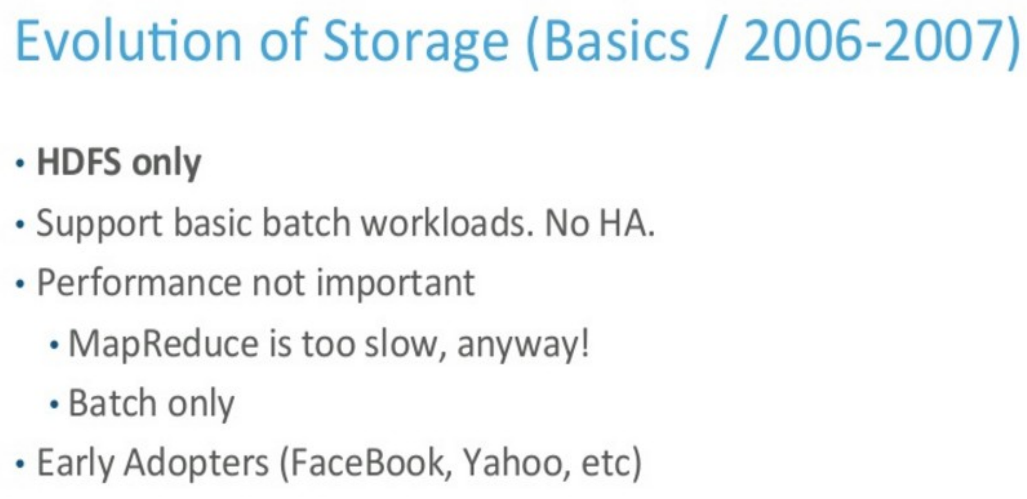
HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (providing BigTable-like capabilities for Hadoop). That is, it provides a fault-tolerant way of storing large quantities of sparse data.

HBase Components

- Region
  - Row of table are stores
- Region Server
  - Hosts the tables
- Master
  - Coordinating the Region Servers
- ZooKeeper
- HDFS
- API
  - The Java Client API



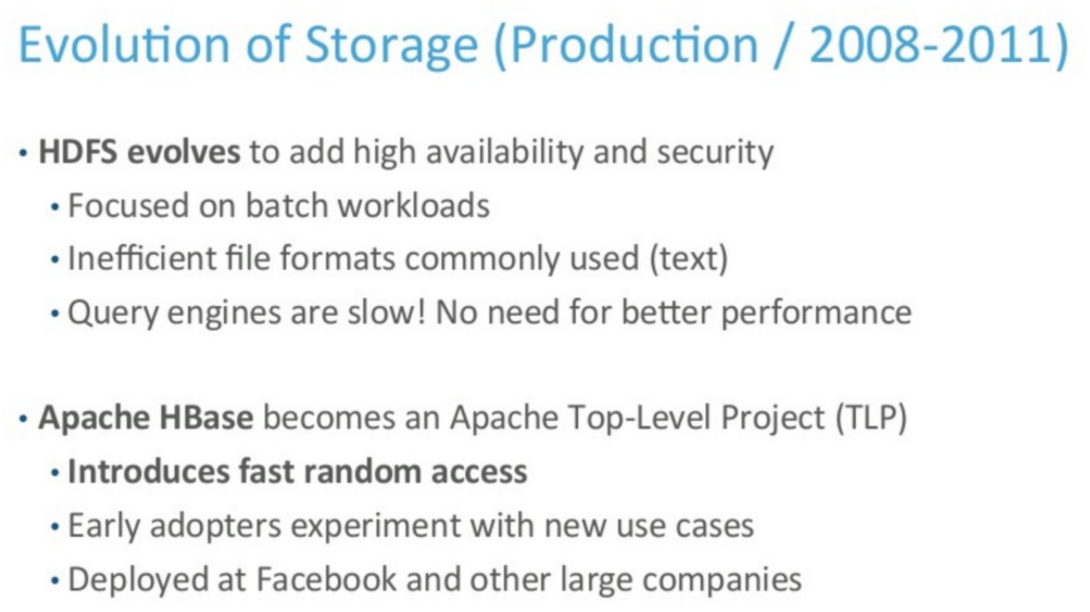
Hive.apache.org



Evolution of Storage (Basics / 2006-2007)

- **HDFS only**
- Support basic batch workloads. No HA.
- Performance not important
  - MapReduce is too slow, anyway!
  - Batch only
- Early Adopters (FaceBook, Yahoo, etc)

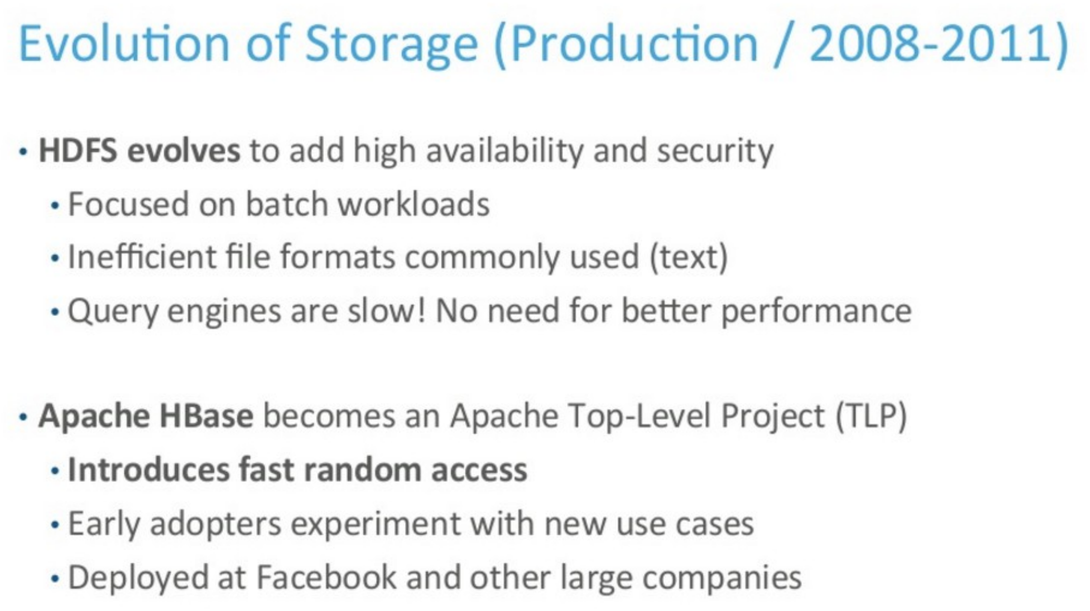
Source: The evolution and future of Hadoop storage: Cloudera



Evolution of Storage (Production / 2008-2011)

- **HDFS evolves** to add high availability and security
  - Focused on batch workloads
  - Inefficient file formats commonly used (text)
  - Query engines are slow! No need for better performance
- **Apache HBase** becomes an Apache Top-Level Project (TLP)
  - **Introduces fast random access**
  - Early adopters experiment with new use cases
  - Deployed at Facebook and other large companies

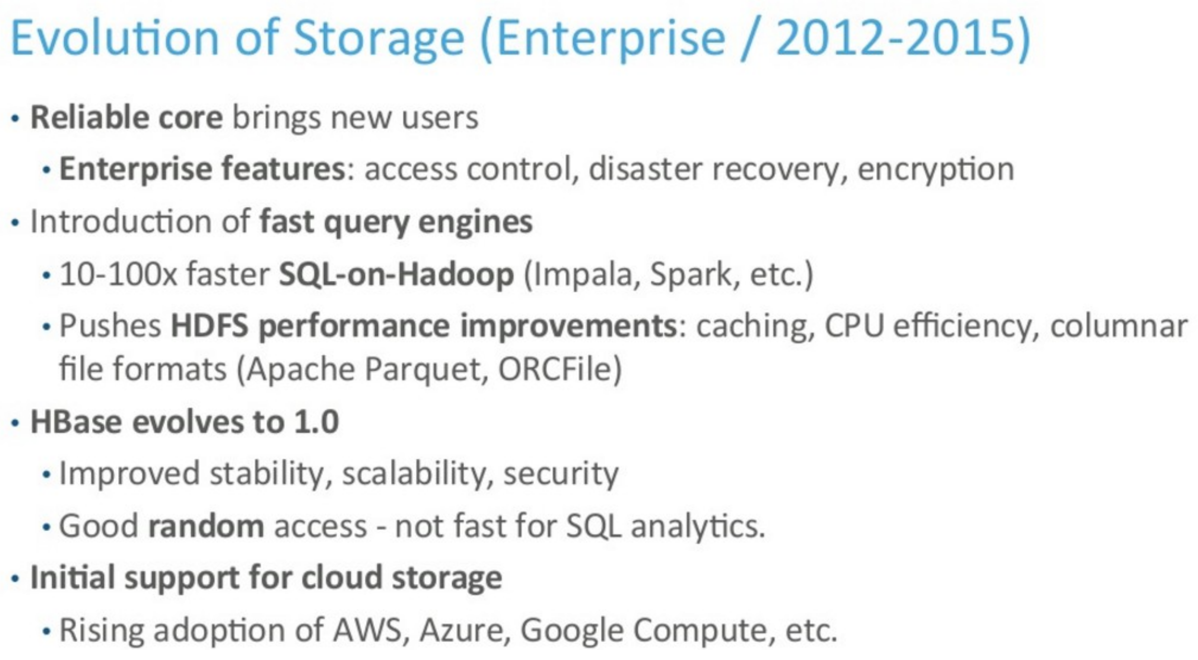
Source: The evolution and future of Hadoop storage: Cloudera



Evolution of Storage (Production / 2008-2011)

- **HDFS evolves** to add high availability and security
  - Focused on batch workloads
  - Inefficient file formats commonly used (text)
  - Query engines are slow! No need for better performance
- **Apache HBase** becomes an Apache Top-Level Project (TLP)
  - **Introduces fast random access**
  - Early adopters experiment with new use cases
  - Deployed at Facebook and other large companies

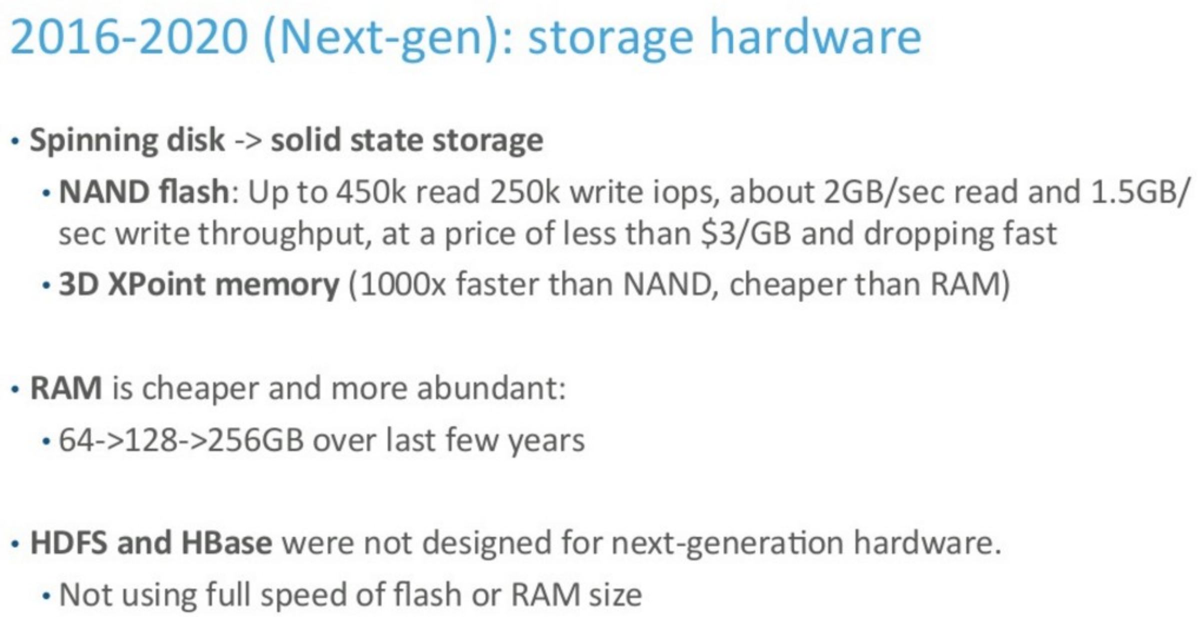
Source: The evolution and future of Hadoop storage: Cloudera



Evolution of Storage (Enterprise / 2012-2015)

- **Reliable core** brings new users
  - **Enterprise features**: access control, disaster recovery, encryption
- Introduction of **fast query engines**
  - 10-100x faster **SQL-on-Hadoop** (Impala, Spark, etc.)
  - Pushes **HDFS performance improvements**: caching, CPU efficiency, columnar file formats (Apache Parquet, ORCFile)
- **HBase evolves to 1.0**
  - Improved stability, scalability, security
  - Good **random** access - not fast for SQL analytics.
- **Initial support for cloud storage**
  - Rising adoption of AWS, Azure, Google Compute, etc.

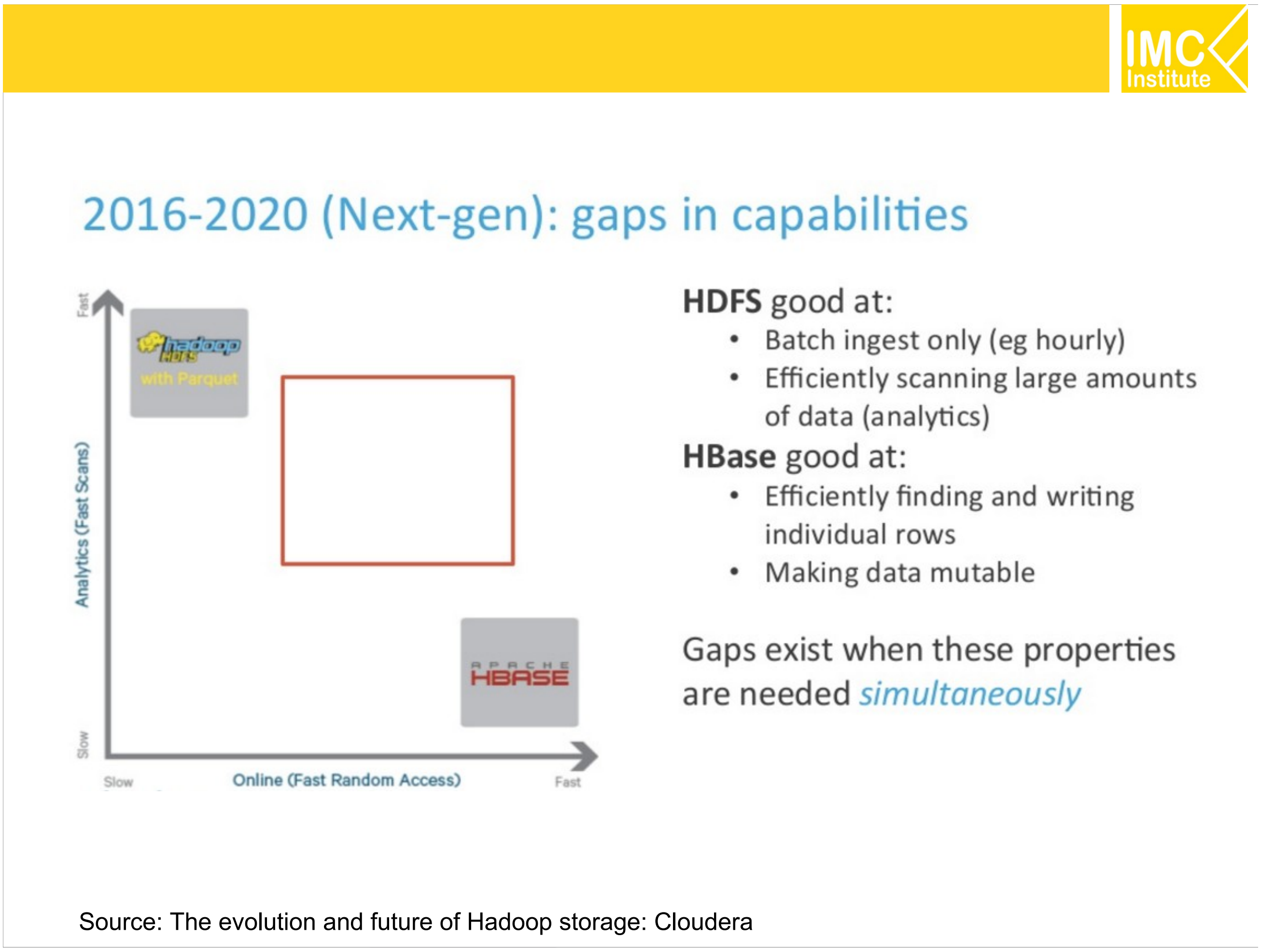
Source: The evolution and future of Hadoop storage: Cloudera



2016-2020 (Next-gen): storage hardware

- **Spinning disk** -> **solid state storage**
  - **NAND flash**: Up to 450k read 250k write iops, about 2GB/sec read and 1.5GB/sec write throughput, at a price of less than $3/GB and dropping fast
  - **3D XPoint memory** (1000x faster than NAND, cheaper than RAM)
- **RAM** is cheaper and more abundant:
  - 64->128->256GB over last few years
- **HDFS and HBase** were not designed for next-generation hardware.
  - Not using full speed of flash or RAM size

Source: The evolution and future of Hadoop storage: Cloudera



2016-2020 (Next-gen): gaps in capabilities

HDFS good at:

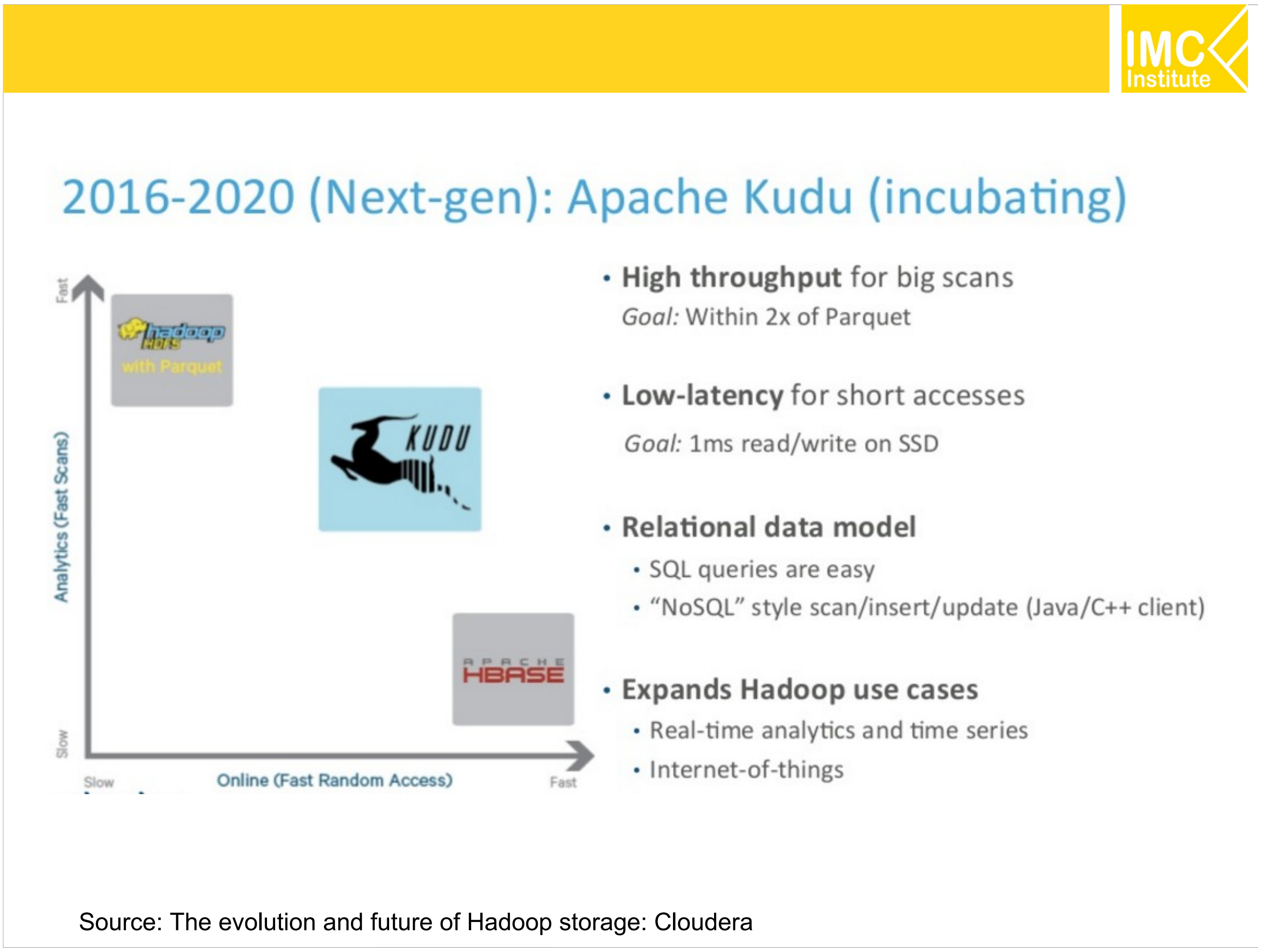
- Batch ingest only (eg hourly)
- Efficiently scanning large amounts of data (analytics)

HBase good at:

- Efficiently finding and writing individual rows
- Making data mutable

Gaps exist when these properties are needed *simultaneously*

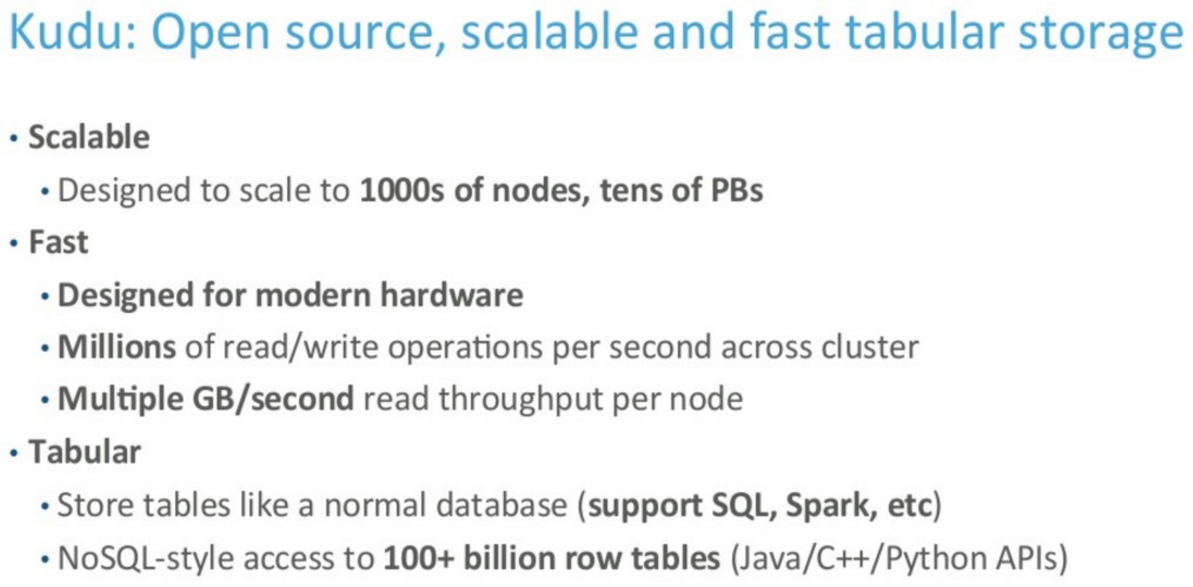
Source: The evolution and future of Hadoop storage: Cloudera



2016-2020 (Next-gen): Apache Kudu (incubating)

- **High throughput** for big scans
  Goal: Within 2x of Parquet
- **Low-latency** for short accesses
  Goal: 1ms read/write on SSD
- **Relational data model**
  - SQL queries are easy
  - "NoSQL" style scan/insert/update (Java/C++ client)
- **Expands Hadoop use cases**
  - Real-time analytics and time series
  - Internet-of-things

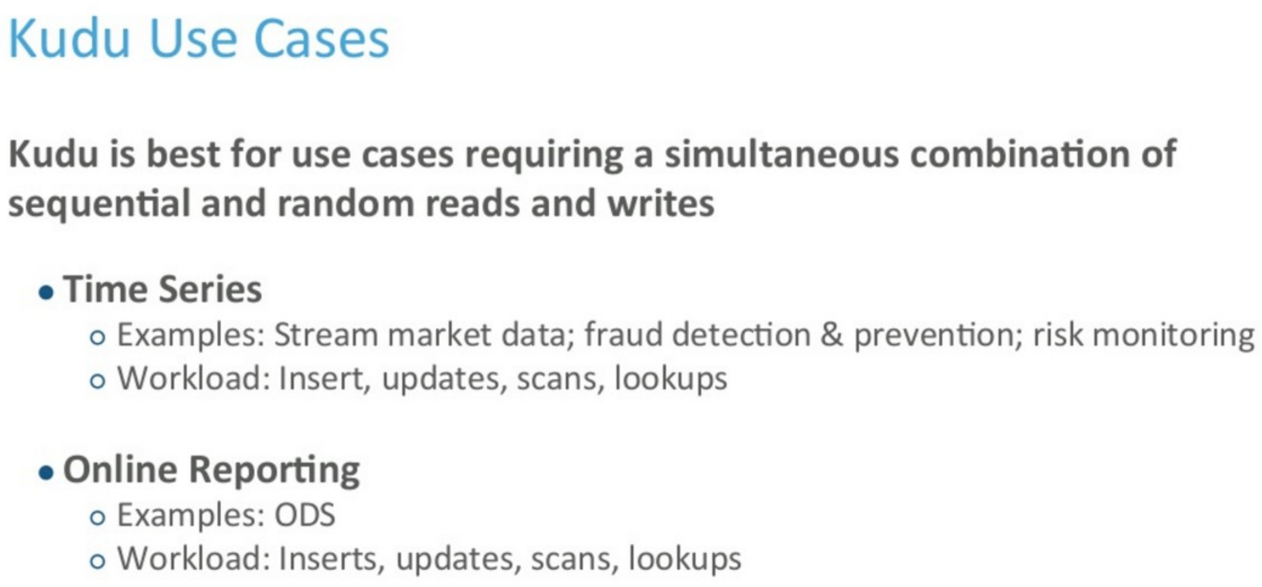
Source: The evolution and future of Hadoop storage: Cloudera



Kudu: Open source, scalable and fast tabular storage

- Scalable
  - Designed to scale to **1000s of nodes, tens of PBs**
- Fast
  - **Designed for modern hardware**
  - **Millions** of read/write operations per second across cluster
  - **Multiple GB/second** read throughput per node
- Tabular
  - Store tables like a normal database (**support SQL, Spark, etc**)
  - NoSQL-style access to **100+ billion row tables** (Java/C++/Python APIs)

Source: The evolution and future of Hadoop storage: Cloudera



Kudu Use Cases

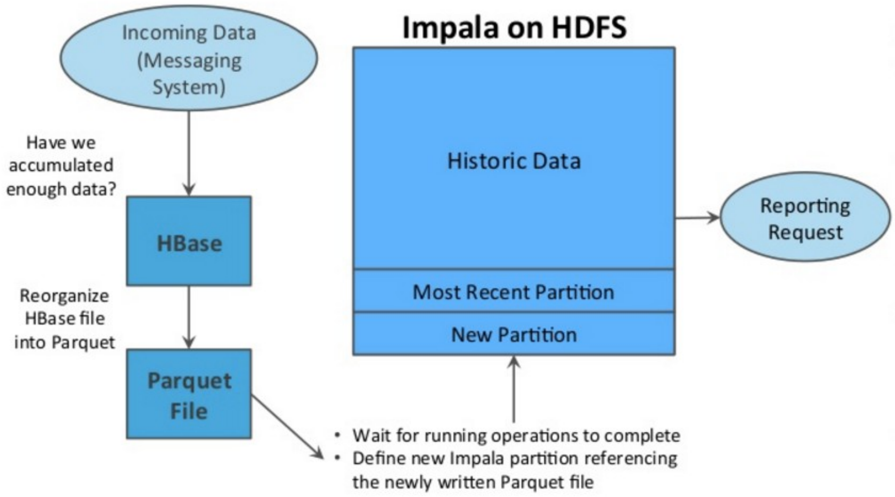
Kudu is best for use cases requiring a simultaneous combination of sequential and random reads and writes

- **Time Series**
  - Examples: Stream market data; fraud detection & prevention; risk monitoring
  - Workload: Insert, updates, scans, lookups
- **Online Reporting**
  - Examples: ODS
  - Workload: Inserts, updates, scans, lookups

Source: Apache Kadu: Fast Analytics and Fast Data: Cloudera

Real-Time Analytics in Hadoop Today

Fraud Detection in the Real World = Storage Complexity



Incoming Data (Messaging System)

Have we accumulated enough data?

HBase

Reorganize HBase file into Parquet

Parquet File

Impala on HDFS

Historic Data

Most Recent Partition

New Partition

Reporting Request

- Wait for running operations to complete
- Define new Impala partition referencing the newly written Parquet file

Considerations:

- How do I handle failure during this process?
- How often do I reorganize data streaming in into a format appropriate for reporting?
- When reporting, how do I see data that has not yet been reorganized?
- How do I ensure that important jobs aren't interrupted by maintenance?

Source: Apache Kadu: Fast Analytics and Fast Data: Cloudera

2016-2020 (Next gen): Predictions

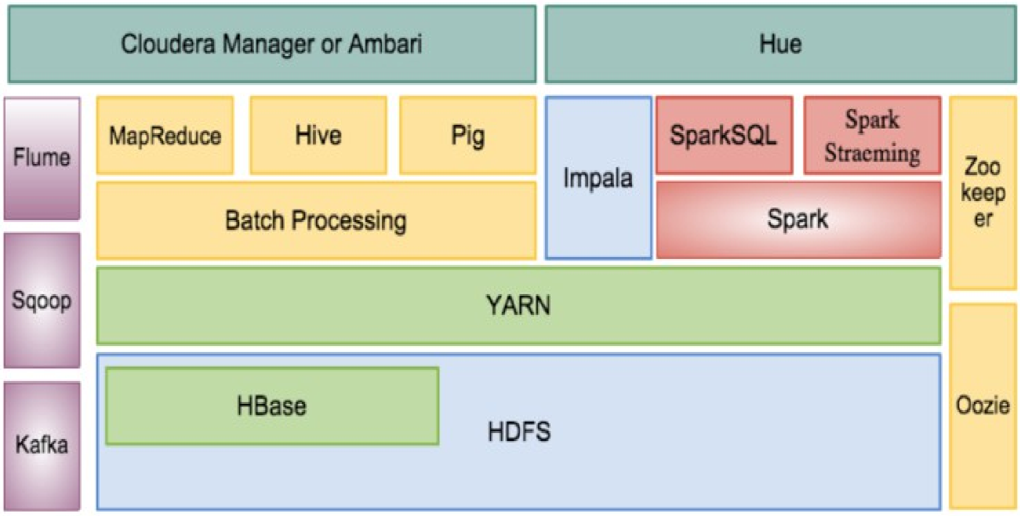
- **Kudu** will evolve an enterprise feature set and enable simple high-performance real-time architectures
  - Increasing ability to migrate traditional applications
- **HDFS and HBase** will continue to innovate and adapt to next generation hardware
  - Steady improvements in **performance, efficiency, and scalability** (e.g. erasure coding)
- **Cloud storage** will become increasingly important
  - **Hadoop ecosystem** will evolve to coexist

Source: The evolution and future of Hadoop storage: Cloudera

Other Hadoop Services

- **Flume:** Collection and import of log and event data
- **Kafka**: Publish-subscribe messaging rethought as a distributed commit log. Fast.
- **Sqoop**: Imports data from relational databases
- **Oozie**: Hadoop workflow.
- **Mahout**: Machine Learning Libray
- **Zookeeper**: Centralized service for maintaining configuration
- **Ambari**: Cluster management

Hadoop Ecosystem



Cloudera Manager or Ambari

Hue

Flume

Sqoop

Kafka

MapReduce

Hive

Pig

Batch Processing

Impala

SparkSQL

Spark Streaming

Spark

Zoo keep er

YARN

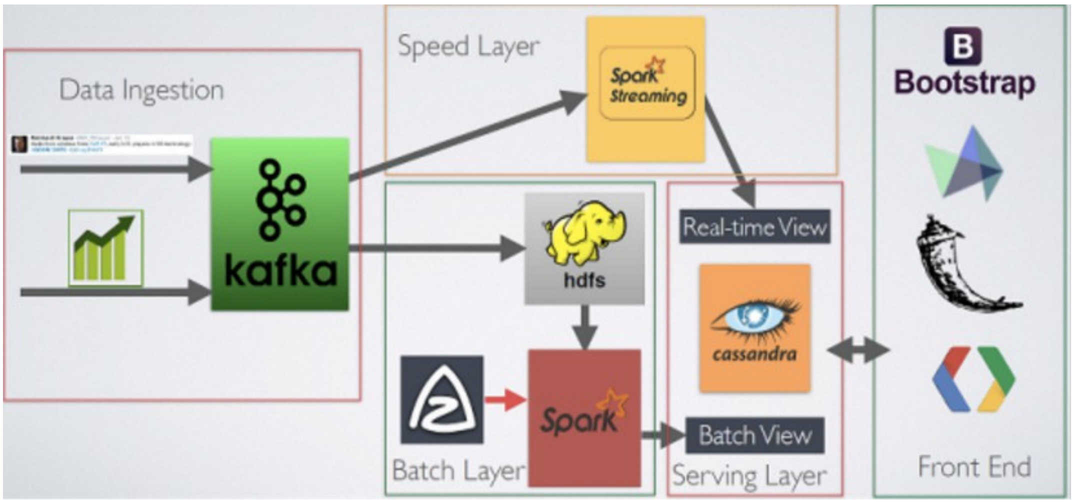
HBase

HDFS

Oozie

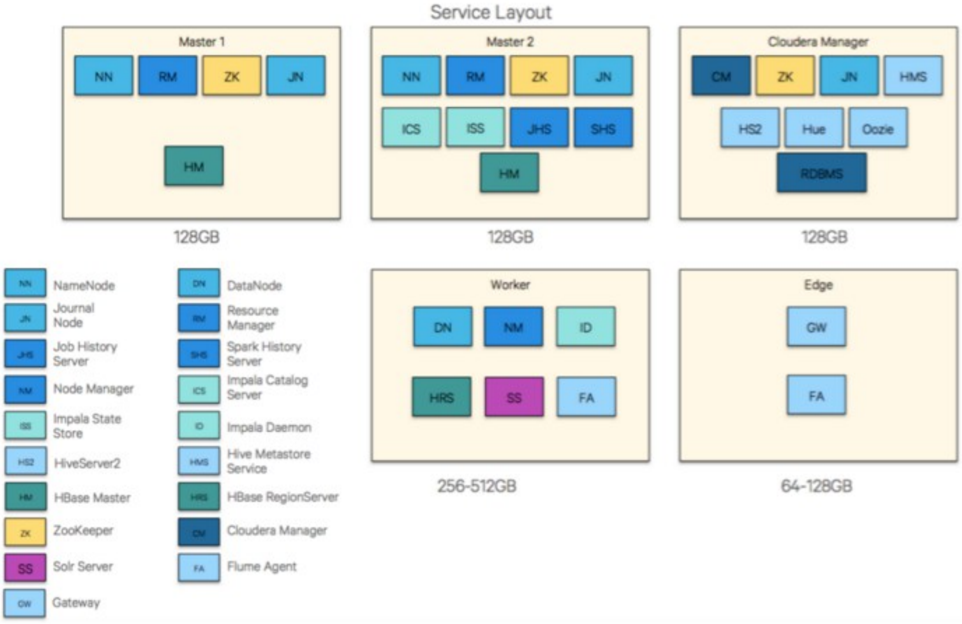
Use Case

Lamda Arch.



Source: http://www.insightdataengineering.com/

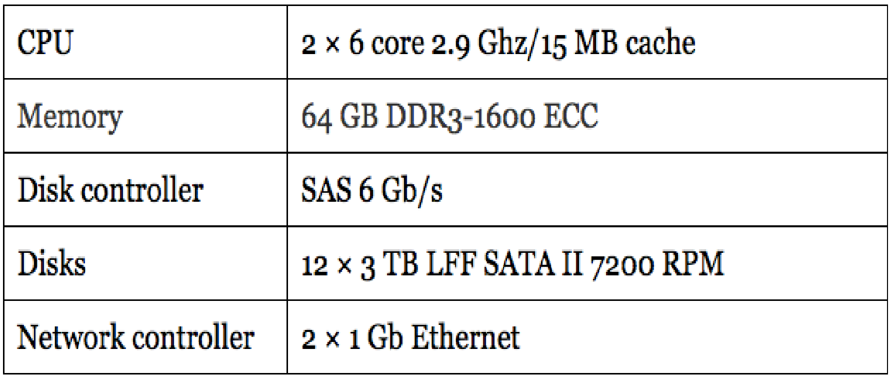
Hadoop Cluster



NN	NameNode	DN	DataNode
JN	Journal Node	RM	Resource Manager
JHS	Job History Server	SHS	Spark History Server
NM	Node Manager	ICS	Impala Catalog Server
ISS	Impala State Store	ID	Impala Daemon
HS2	HiveServer2	HMS	Hive Metastore Service
HM	HBase Master	HRS	HBase RegionServer
ZK	ZooKeeper	CM	Cloudera Manager
SS	Solr Server	FA	Flume Agent
GW	Gateway		

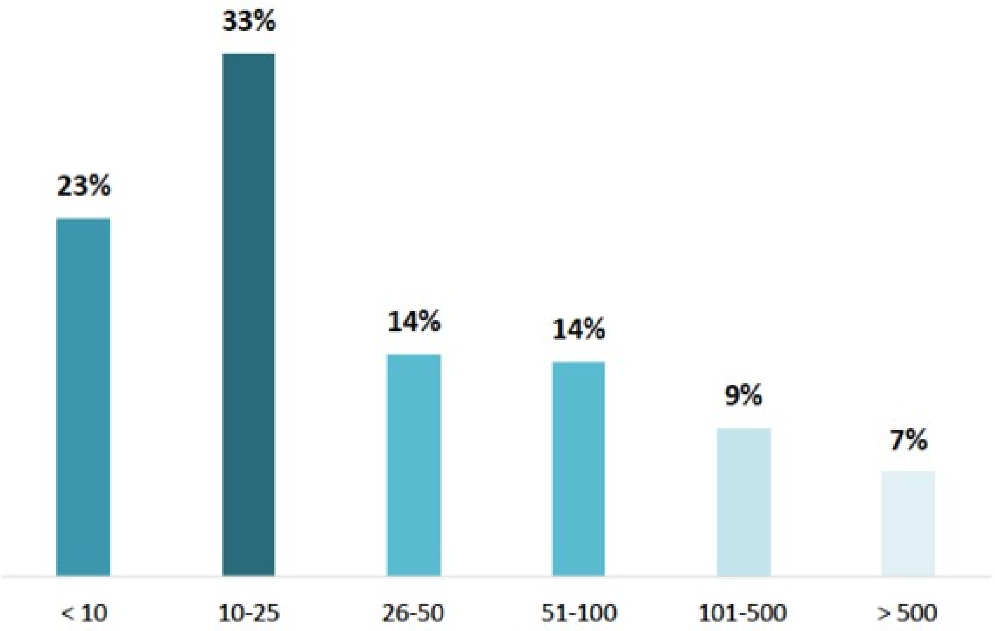
Source: How-to: Deploy Apache Hadoop Clusters Like a Boss, Jeff Holoman

Hadoop Hardware Sizing



CPU	2 × 6 core 2.9 Ghz/15 MB cache
Memory	64 GB DDR3-1600 ECC
Disk controller	SAS 6 Gb/s
Disks	12 × 3 TB LFF SATA II 7200 RPM
Network controller	2 × 1 Gb Ethernet

Hadoop Cluster Size: Survey



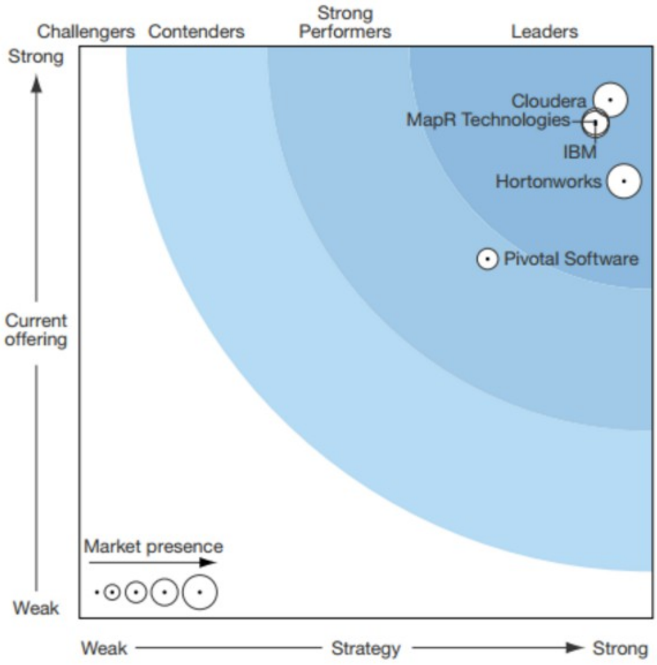
Source: AtScale

Hadoop Distribution

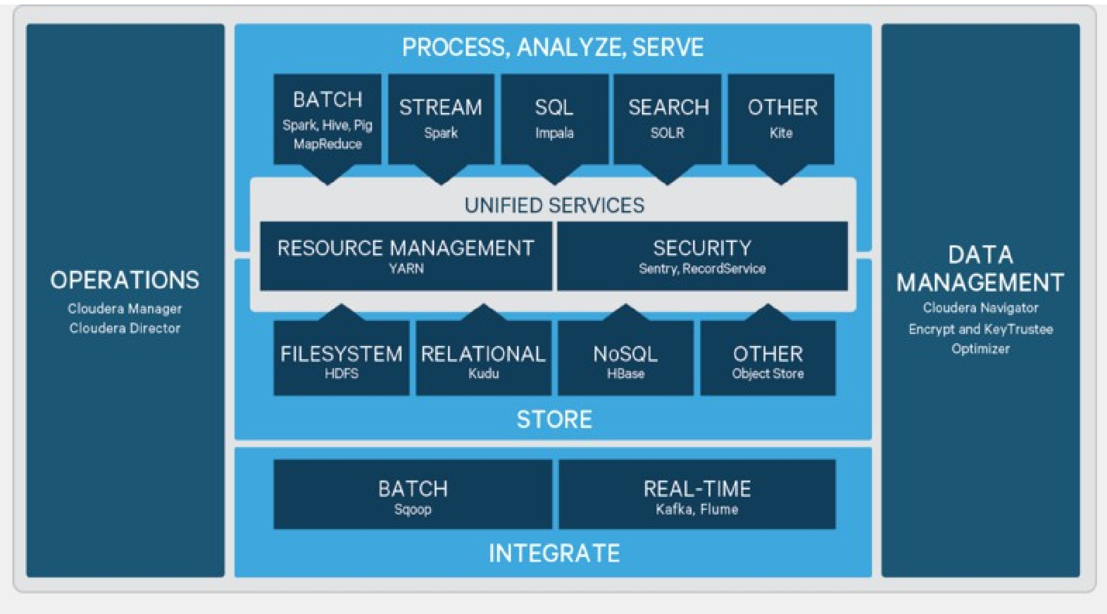
powered by Apache
hadoop
MAPR
cloudera
Hortonworks
Pivotal
IBM InfoSphere BigInsights
amazon web services

Azure HDInsight

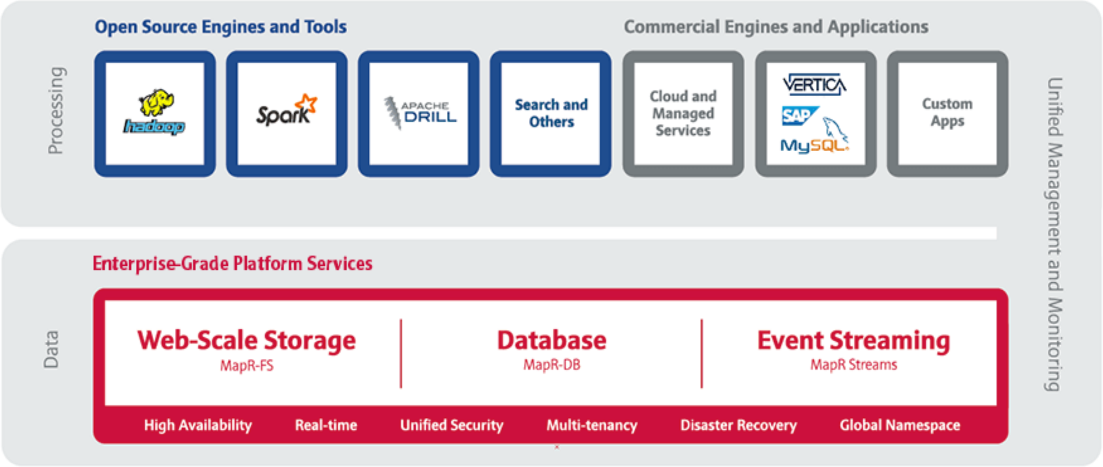
The Forrester Wave™: Big Data Hadoop Distributions: Q1 2006



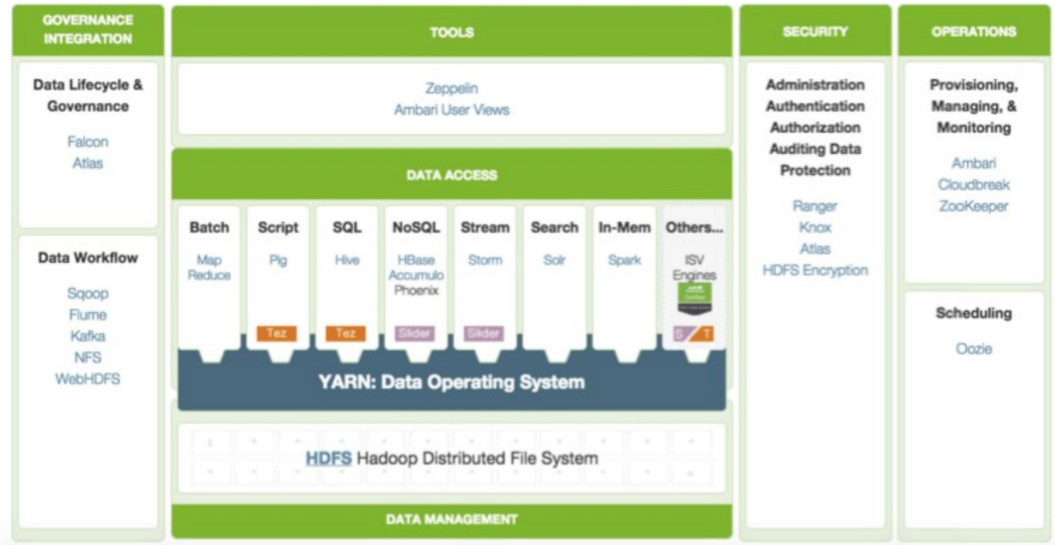
Cloudera



MapR



Hortonworks



GOVERNANCE INTEGRATION

Data Lifecycle & Governance

Falcon
Atlas

Data Workflow

Sqoop
Flume
Kafka
NFS
WebHDFS

TOOLS

Zeppelin
Ambari User Views

DATA ACCESS

Batch	Script	SQL	NoSQL	Stream	Search	In-Mem	Others...
Map Reduce	Pig	Hive	HBase Accumulo Phoenix	Storm	Solr	Spark	ISV Engines
	Tez	Tez	Slider	Slider			S T

YARN: Data Operating System

HDFS Hadoop Distributed File System

DATA MANAGEMENT

SECURITY

Administration
Authentication
Authorization
Auditing Data
Protection

Ranger
Knox
Atlas
HDFS Encryption

OPERATIONS

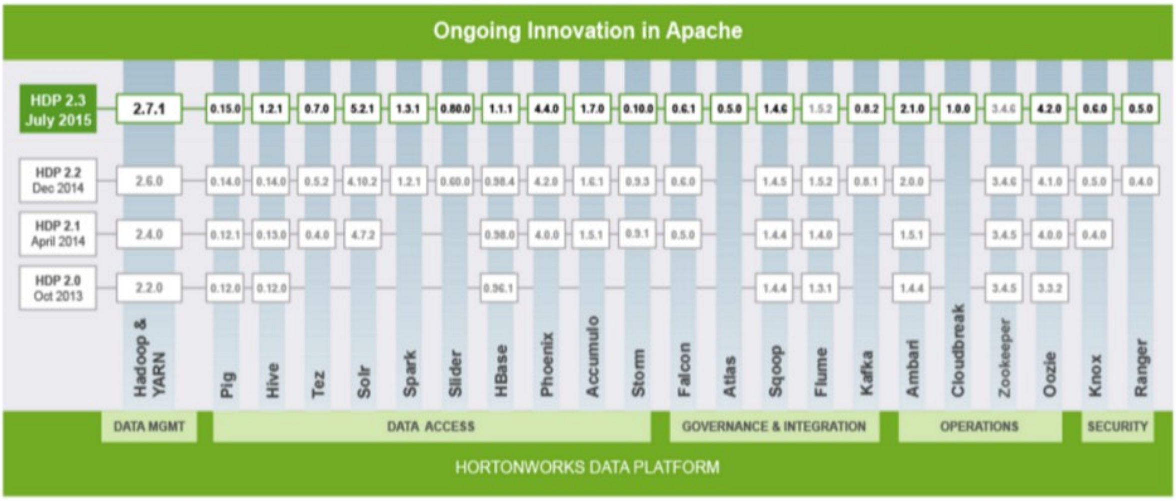
Provisioning, Managing, & Monitoring

Ambari
Cloudbreak
ZooKeeper

Scheduling

Oozie

Hortonworks



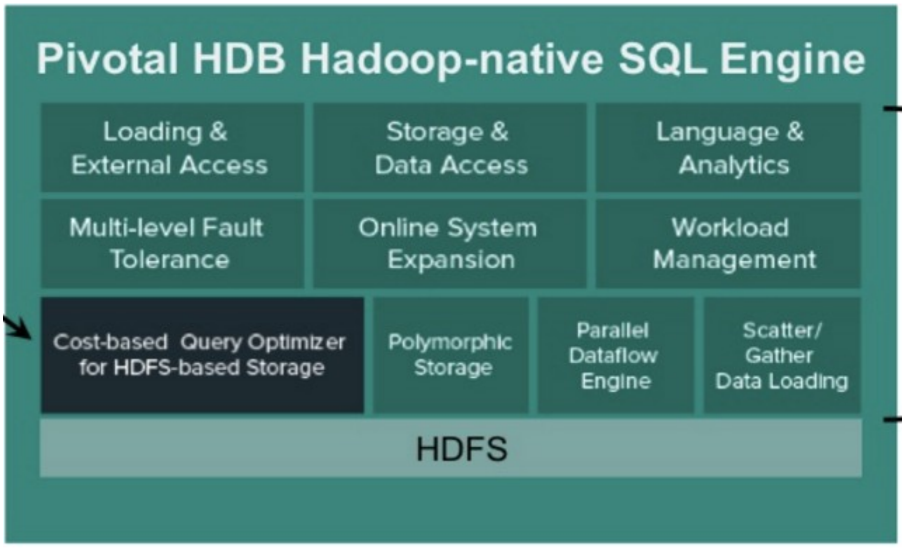
Ongoing Innovation in Apache

	DATA MGMT	DATA ACCESS										
	Hadoop & YARN	Pig	Hive	Tez	Solr	Spark	Slider	HBase	Phoenix	Accumulo	Storm
HDP 2.3 July 2015	2.7.1	0.15.0	1.2.1	0.7.0	5.2.1	1.3.1	0.80.0	1.1.1	4.4.0	1.7.0	0.10.0
HDP 2.2 Dec 2014	2.6.0	0.14.0	0.14.0	0.5.2	4.10.2	1.2.1	0.60.0	0.98.4	4.2.0	1.6.1	0.9.3
HDP 2.1 April 2014	2.4.0	0.12.1	0.13.0	0.4.0	4.7.2			0.98.0	4.0.0	1.5.1	0.9.1
HDP 2.0 Oct 2013	2.2.0	0.12.0	0.12.0					0.96.1			

	GOVERNANCE & INTEGRATION					OPERATIONS				SECURITY	
	Falcon	Atlas	Sqoop	Flume	Kafka	Ambari	Cloudbreak	Zookeeper	Oozie	Knox	Ranger
HDP 2.3 July 2015	0.6.1	0.5.0	1.4.6	1.5.2	0.8.2	2.1.0	1.0.0	3.4.6	4.2.0	0.6.0	0.5.0
HDP 2.2 Dec 2014	0.6.0		1.4.5	1.5.2	0.8.1	2.0.0		3.4.6	4.1.0	0.5.0	0.4.0
HDP 2.1 April 2014	0.5.0		1.4.4	1.4.0		1.5.1		3.4.5	4.0.0	0.4.0	
HDP 2.0 Oct 2013			1.4.4	1.3.1		1.4.4		3.4.5	3.3.2		

HORTONWORKS DATA PLATFORM

Pivotal HDB



Pivotal HDB Hadoop-native SQL Engine

Loading & External Access

Storage & Data Access

Language & Analytics

Multi-level Fault Tolerance

Online System Expansion

Workload Management

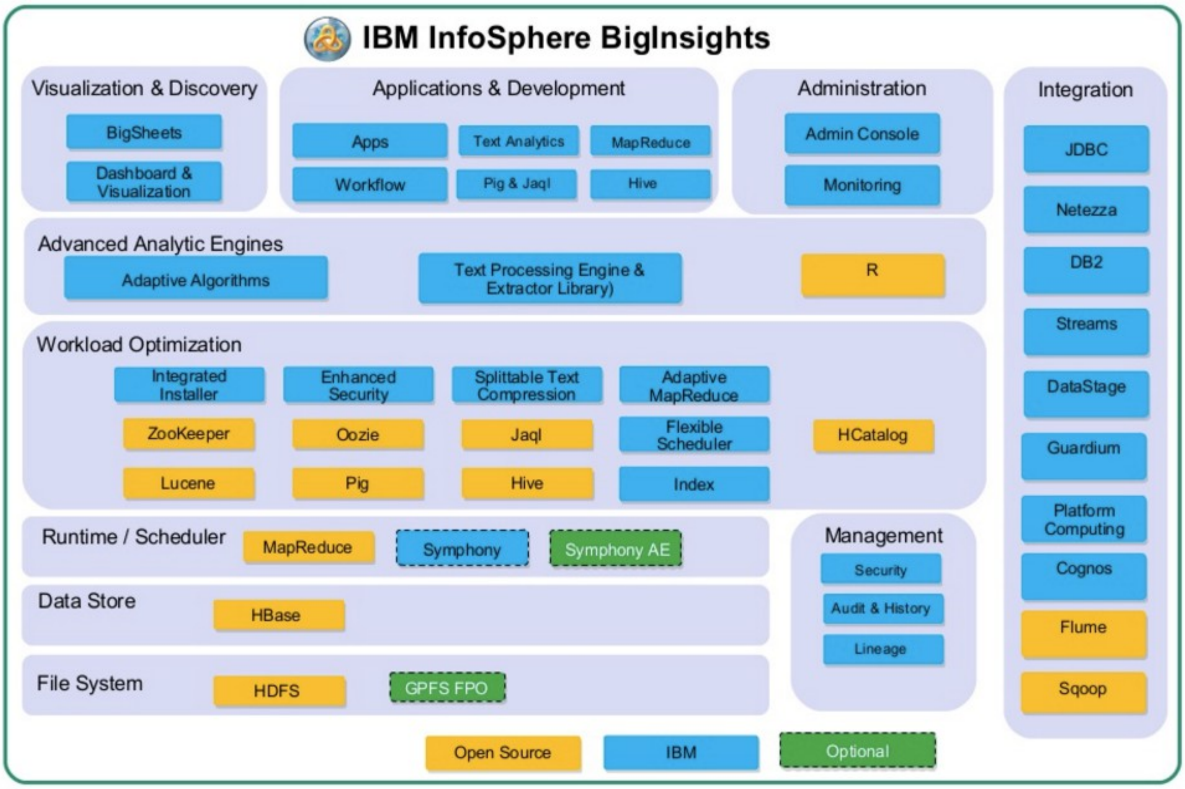
Cost-based Query Optimizer for HDFS-based Storage

Polymorphic Storage

Parallel Dataflow Engine

Scatter/ Gather Data Loading

HDFS



IBM InfoSphere BigInsights

Visualization & Discovery: BigSheets; Dashboard & Visualization

Applications & Development: Apps; Text Analytics; MapReduce; Workflow; Pig & Jaql; Hive

Administration: Admin Console; Monitoring

Integration: JDBC; Netezza; DB2; Streams; DataStage; Guardium; Platform Computing; Cognos; Flume; Sqoop

Advanced Analytic Engines: Adaptive Algorithms; Text Processing Engine & Extractor Library); R

Workload Optimization: Integrated Installer; Enhanced Security; Splittable Text Compression; Adaptive MapReduce; ZooKeeper; Oozie; Jaql; Flexible Scheduler; HCatalog; Lucene; Pig; Hive; Index

Runtime / Scheduler: MapReduce; Symphony; Symphony AE

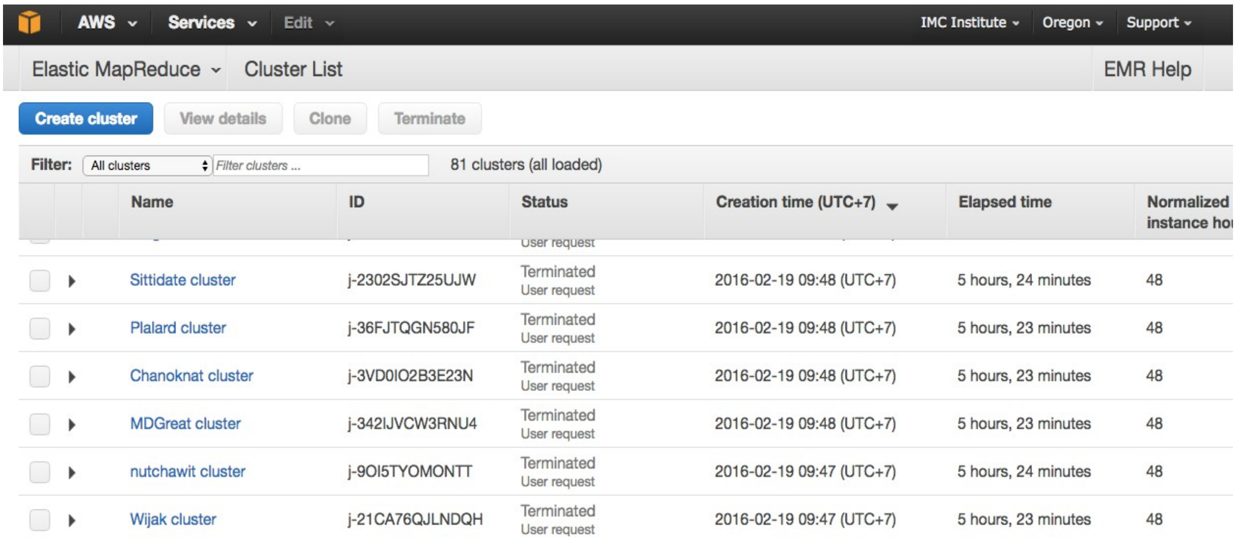
Data Store: HBase

File System: HDFS; GPFS FPO

Management: Security; Audit & History; Lineage

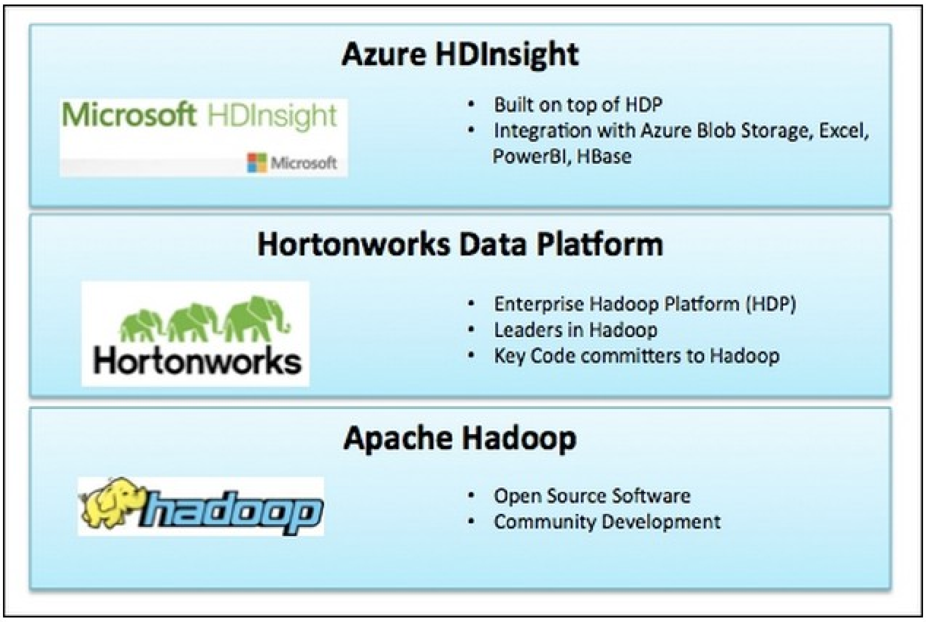
Open Source | IBM | Optional

Amazon EMR



81 clusters (all loaded)

Name	ID	Status	Creation time (UTC+7)	Elapsed time	Normalized instance ho
		User request			
Sittidate cluster	j-2302SJTZ25UJW	Terminated User request	2016-02-19 09:48 (UTC+7)	5 hours, 24 minutes	48
Plalard cluster	j-36FJTQGN580JF	Terminated User request	2016-02-19 09:48 (UTC+7)	5 hours, 23 minutes	48
Chanoknat cluster	j-3VD0IO2B3E23N	Terminated User request	2016-02-19 09:48 (UTC+7)	5 hours, 23 minutes	48
MDGreat cluster	j-342IJVCW3RNU4	Terminated User request	2016-02-19 09:48 (UTC+7)	5 hours, 23 minutes	48
nutchawit cluster	j-9OI5TYOMONTT	Terminated User request	2016-02-19 09:47 (UTC+7)	5 hours, 24 minutes	48
Wijak cluster	j-21CA76QJLNDQH	Terminated User request	2016-02-19 09:47 (UTC+7)	5 hours, 23 minutes	48



Azure HDInsight

- Built on top of HDP
- Integration with Azure Blob Storage, Excel, PowerBI, HBase

Hortonworks Data Platform

- Enterprise Hadoop Platform (HDP)
- Leaders in Hadoop
- Key Code committers to Hadoop

Apache Hadoop

- Open Source Software
- Community Development

Source: HDInsight Essentials - Second Edition

www.facebook.com/imcinstitute

Thank you

thanachart@imcinstitute.com

www.imcinstitute.com

www.facebook.com/imcinstitute

www.slideshare.net/imcinstitute

www.thanachart.org