Tamkang University

Big Data Mining

Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem

1071BDM04
TLVXM1A (M2244) (8619) (Fall 2018)
(MBA, DBETKU) (3 Credits, Required) [Full English Course]
(Master's Program in Digital Business and Economics)
Mon, 9, 10, 11, (16:10-19:00) (B206)

Min-Yuh Day, Ph.D.
Assistant Professor
Department of Information Management
Tamkang University

http://mail.tku.edu.tw/myday

2018-10-08

Course Schedule (1/2)

Tamkang University

Week	Date	Subject/Topics
1	2018/09/10	Course Orientation for Big Data Mining
2	2018/09/17	ABC: AI, Big Data, Cloud Computing
3	2018/09/24	Mid-Autumn Festival (Day off)
4	2018/10/01	Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data
5	2018/10/08	Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem
6	2018/10/15	Foundations of Big Data Mining in Python
7	2018/10/22	Supervised Learning: Classification and Prediction
8	2018/10/29	Unsupervised Learning: Cluster Analysis
9	2018/11/05	Unsupervised Learning: Association Analysis

Course Schedule (2/2)

Tamkang University

Week	Date	Subject/Topics
10	2018/11/12	Midterm Project Report
11	2018/11/19	Machine Learning with Scikit-Learn in Python
12	2018/11/26	Deep Learning for Finance Big Data with TensorFlow
13	2018/12/03	Convolutional Neural Networks (CNN)
14	2018/12/10	Recurrent Neural Networks (RNN)
15	2018/12/17	Reinforcement Learning (RL)
16	2018/12/24	Social Network Analysis (SNA)
17	2018/12/31	Bridge Holiday (Extra Day Off)
18	2019/01/07	Final Project Presentation

Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem

National Security | Cyber security | Maritime security | Smarter Transport | ...

VISUAL ANALYTICS

DYNAMIC & INTERACTIVE
Dashboard Graph
Map

ENHANCE
Understanding Investigation
User Experience

BIG ANALYTICS

QUERY & FILTER
Complex queries
R²I²

DETECT
Anomalies
Communities
Typologies

PREDICT
Tending
Real-time
Prediction

DECIDE
Simulation
Optimization

BIG DATA – Batch

BIG DATA – Real Time

DATA

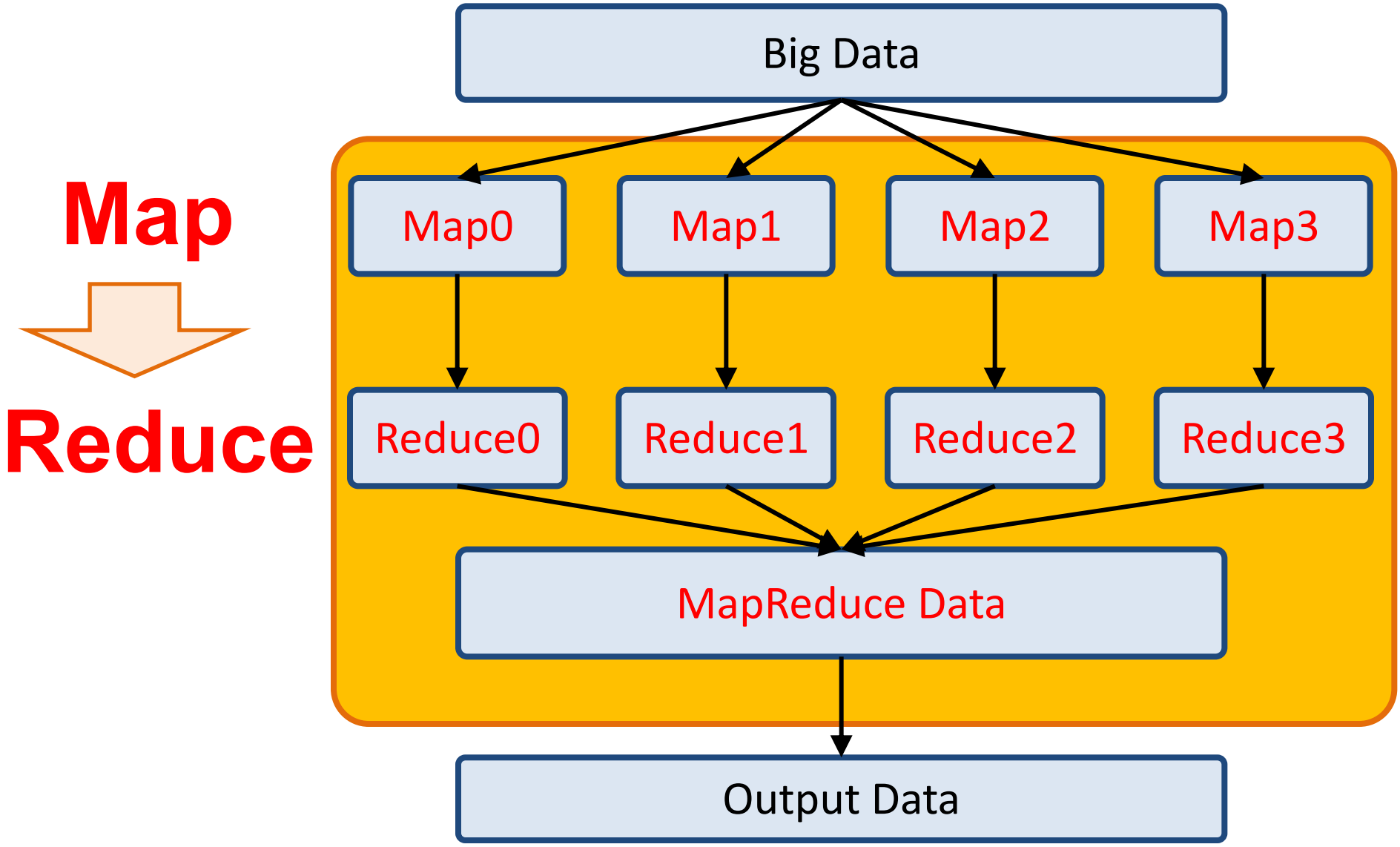
Complex by nature

Complex by structure

Source: https://www.thalesgroup.com/en/worldwide/big-data/big-data-big-analytics-visual-analytics-what-does-it-all-mean

MapReduce Paradigm

MapReduce Paradigm



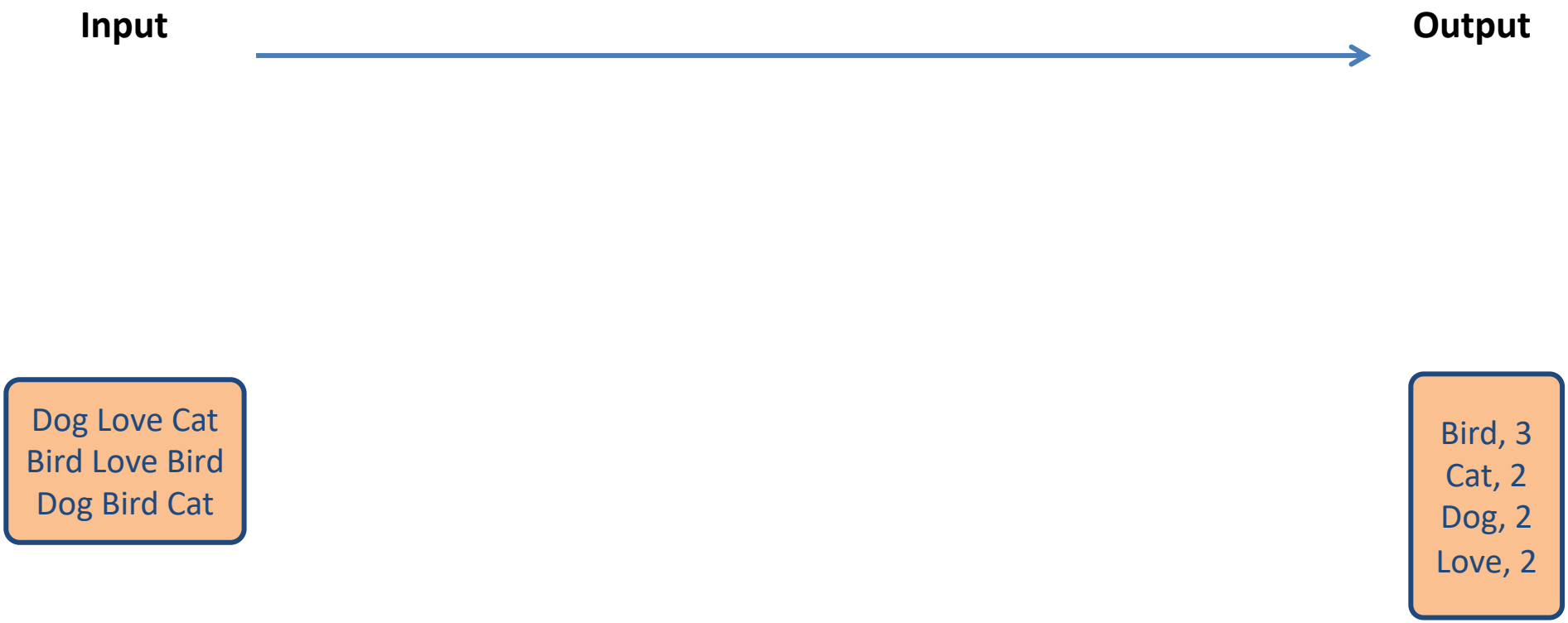
MapReduce Word Count

Input

Dog Love Cat
Bird Love Bird
Dog Bird Cat

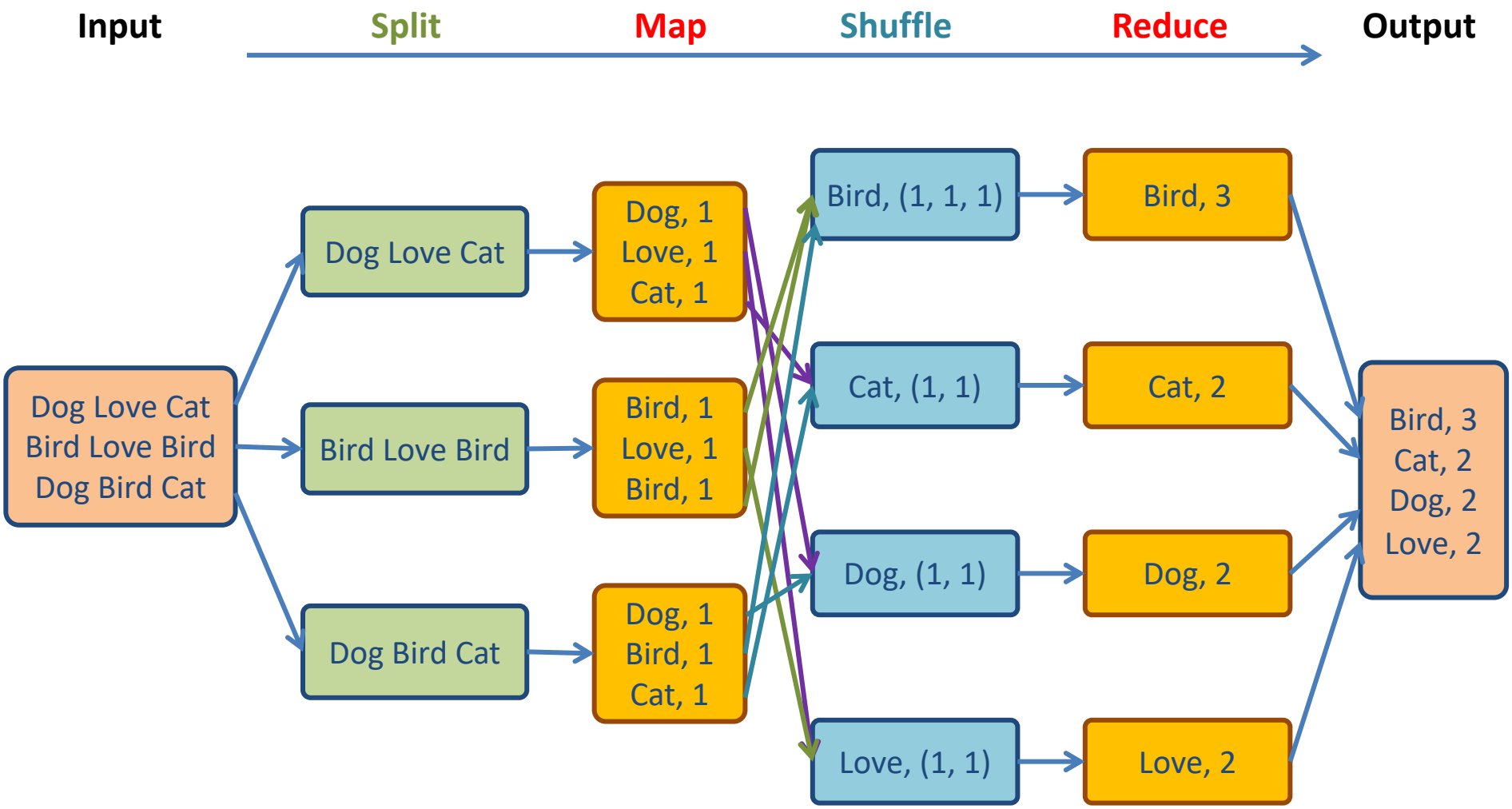
Source: https://www.edureka.co/blog/mapreduce-tutorial/

MapReduce Word Count



Source: https://www.edureka.co/blog/mapreduce-tutorial/

MapReduce Word Count



Source: https://www.edureka.co/blog/mapreduce-tutorial/

Hadoop Ecosystem

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

Source: http://hadoop.apache.org/

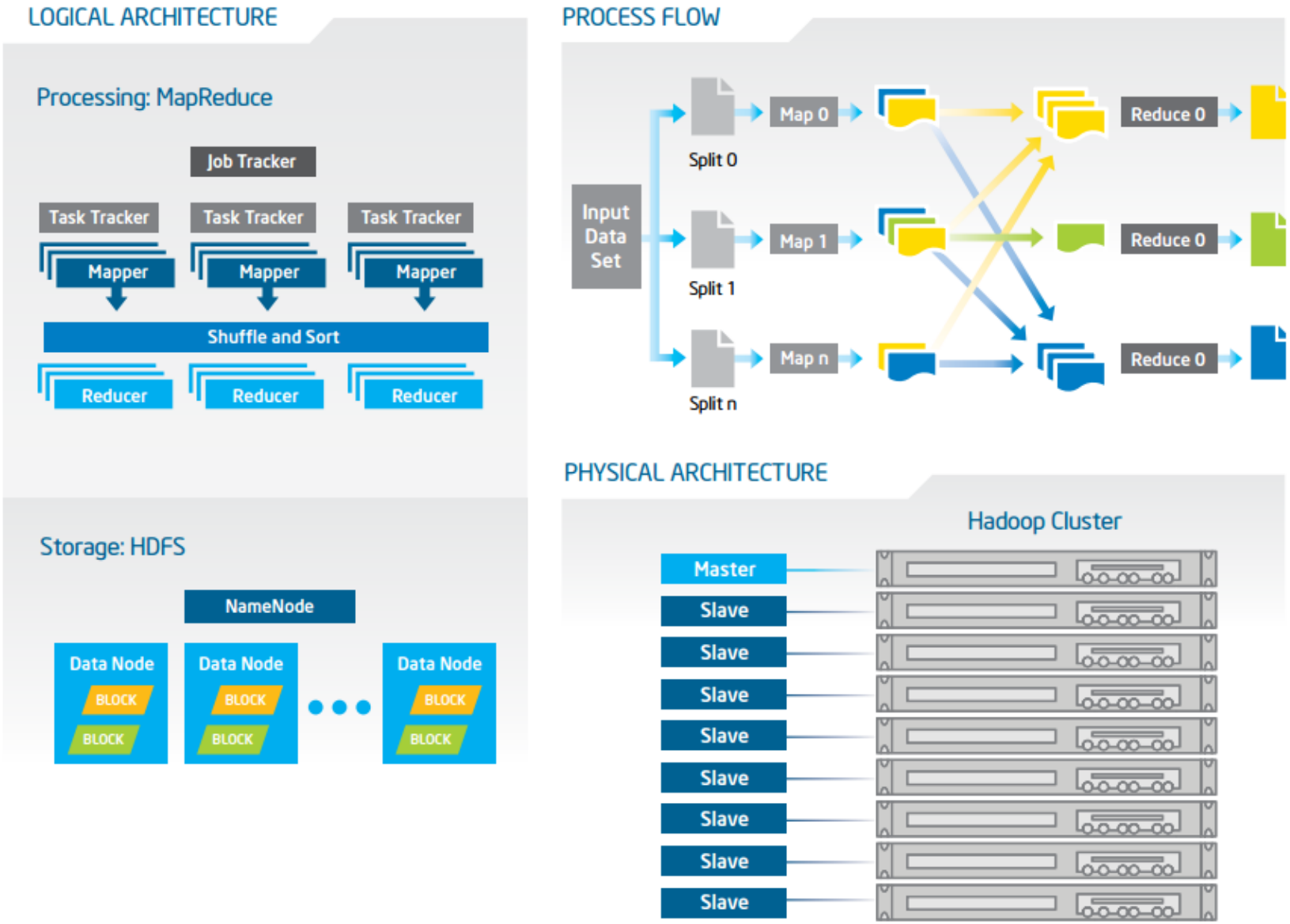
MapReduce Processing

\+

HDFS Storage

Source: http://hadoop.apache.org/

Big Data with Hadoop Architecture

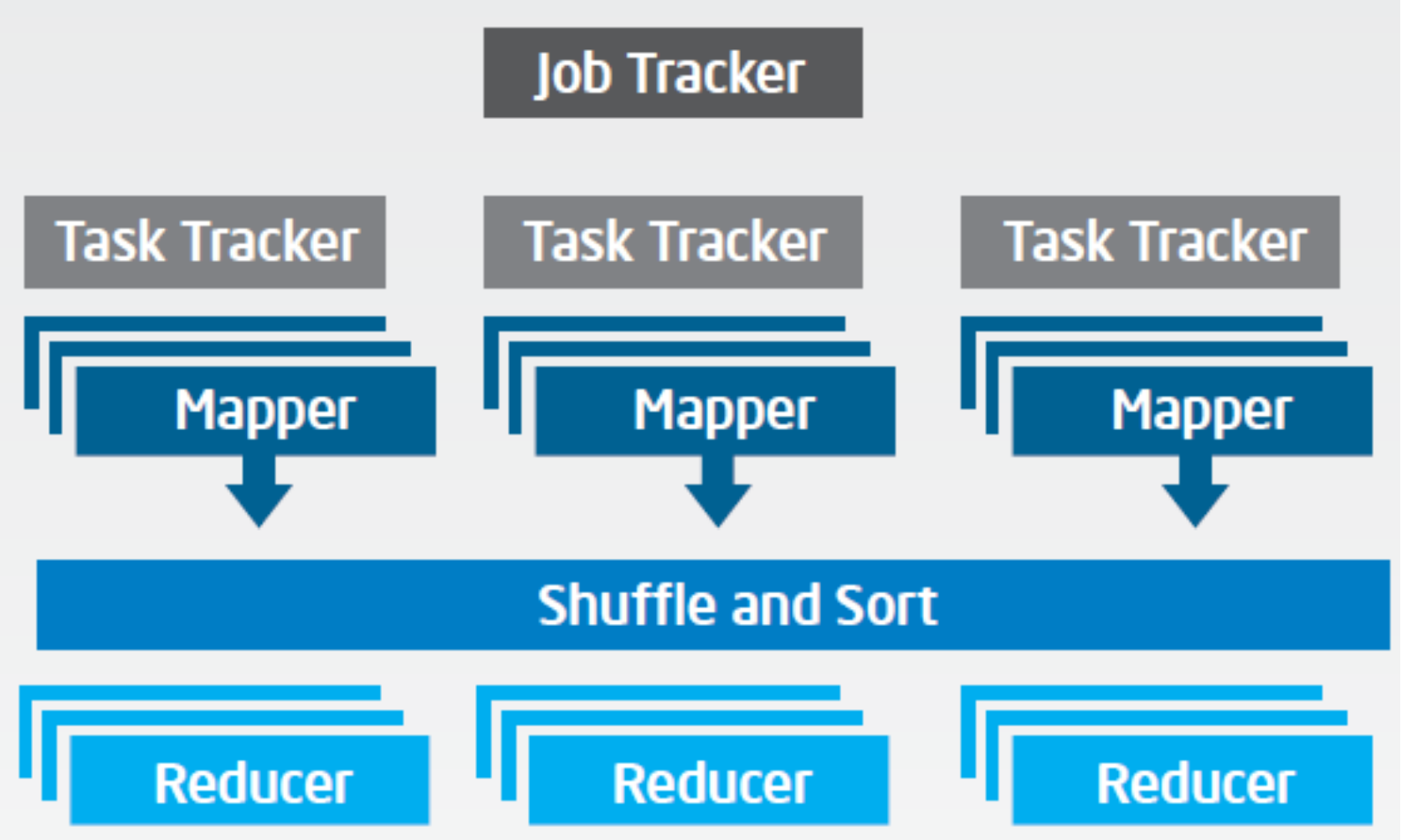


Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Big Data with Hadoop Architecture

Logical Architecture

Processing: MapReduce

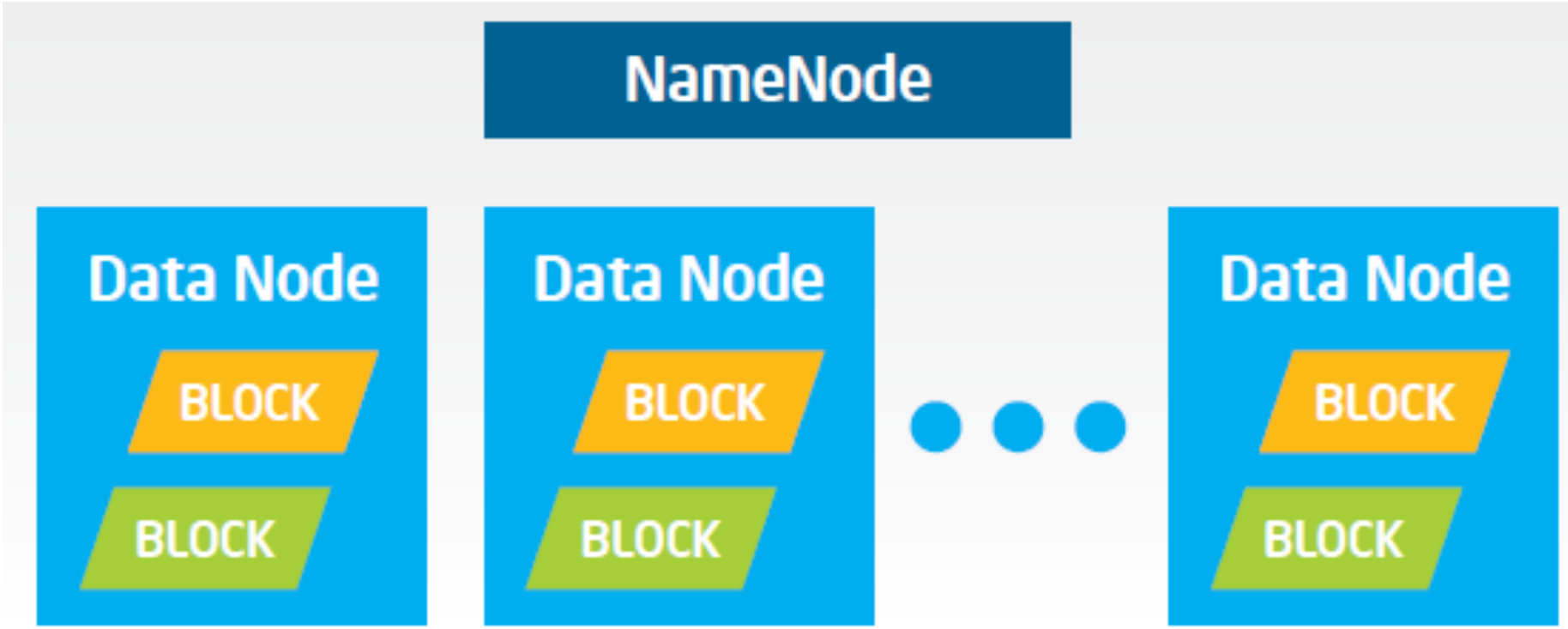


Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Big Data with Hadoop Architecture

Logical Architecture

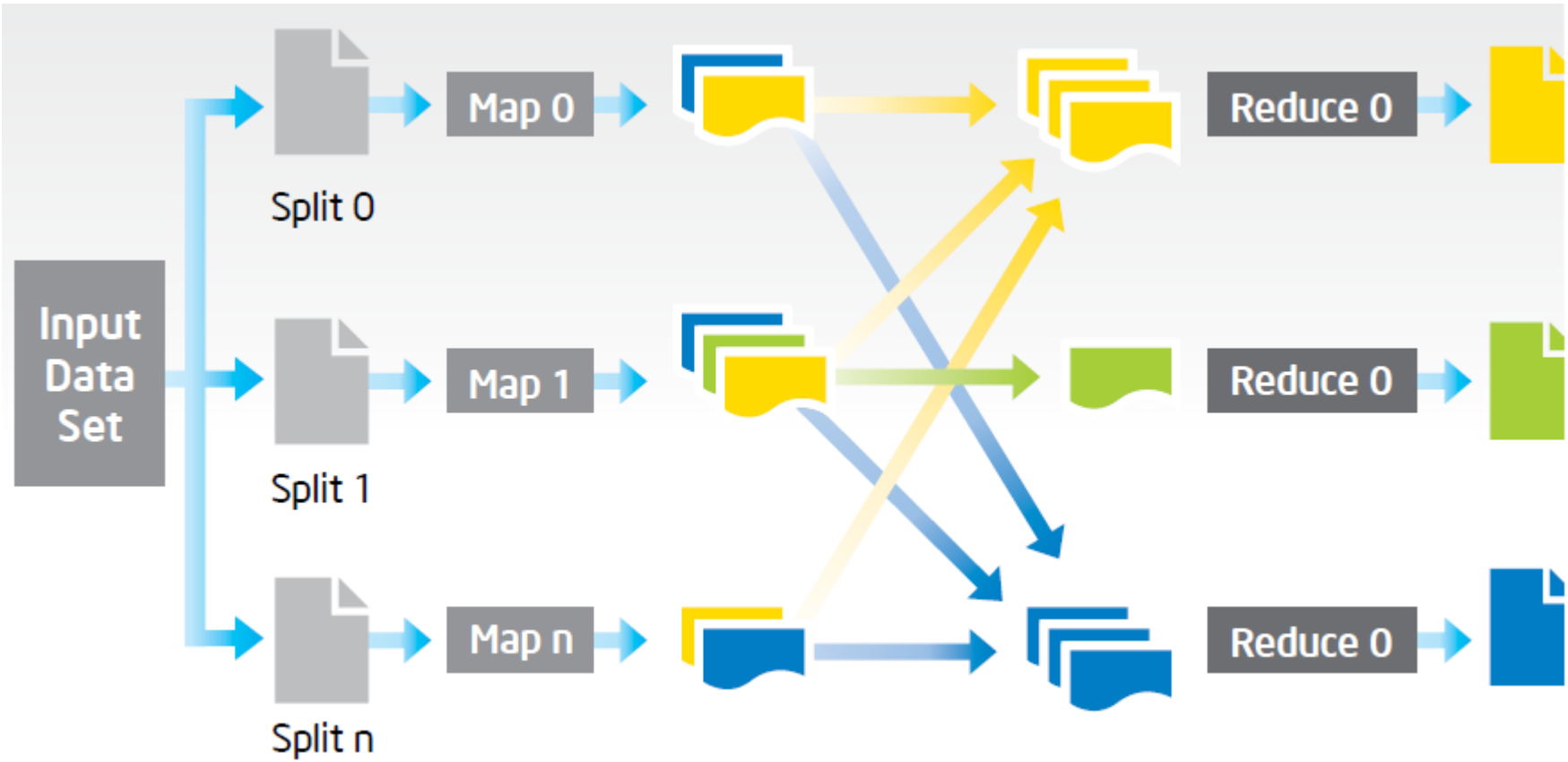
Storage: HDFS



Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Big Data with Hadoop Architecture

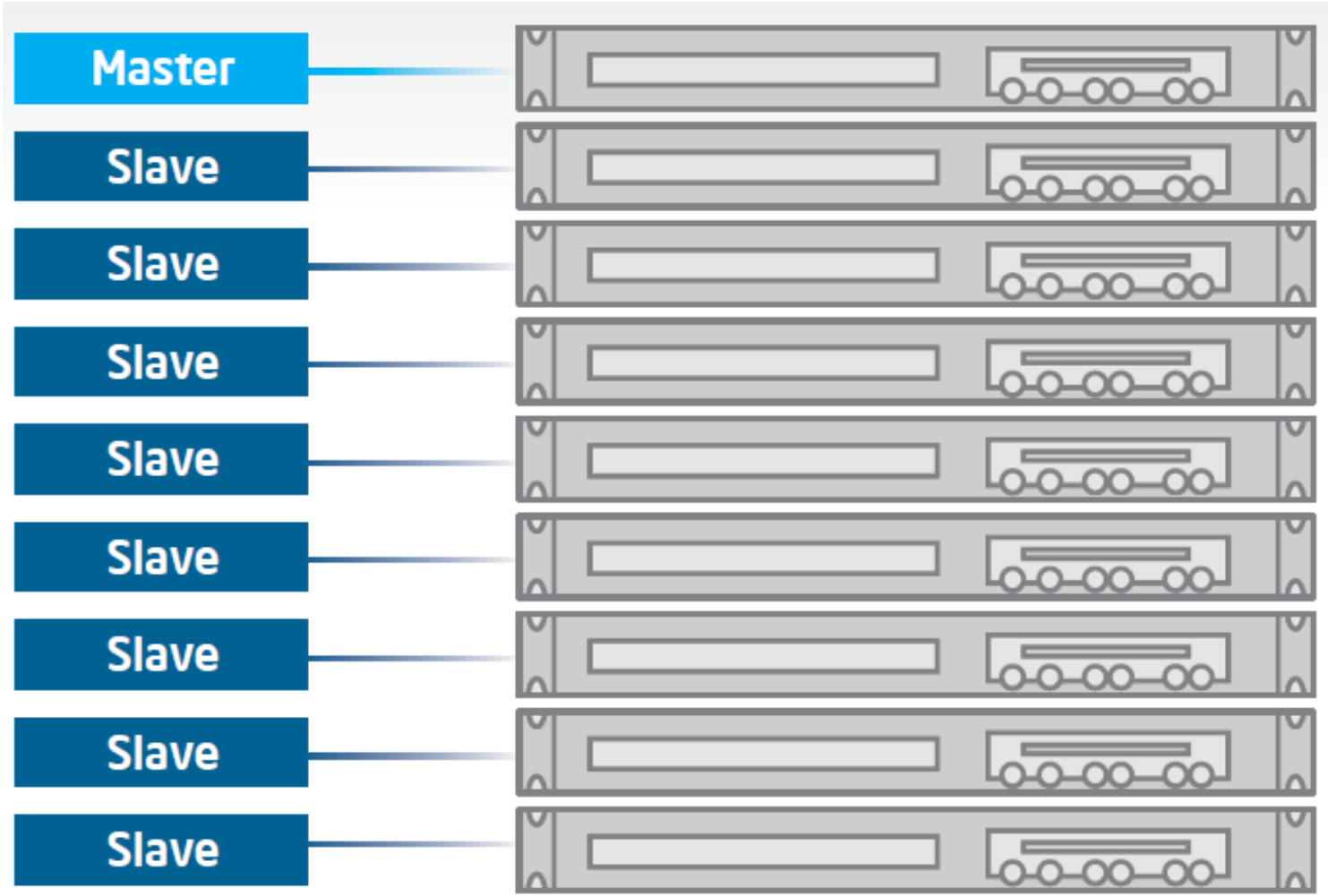
Process Flow



Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

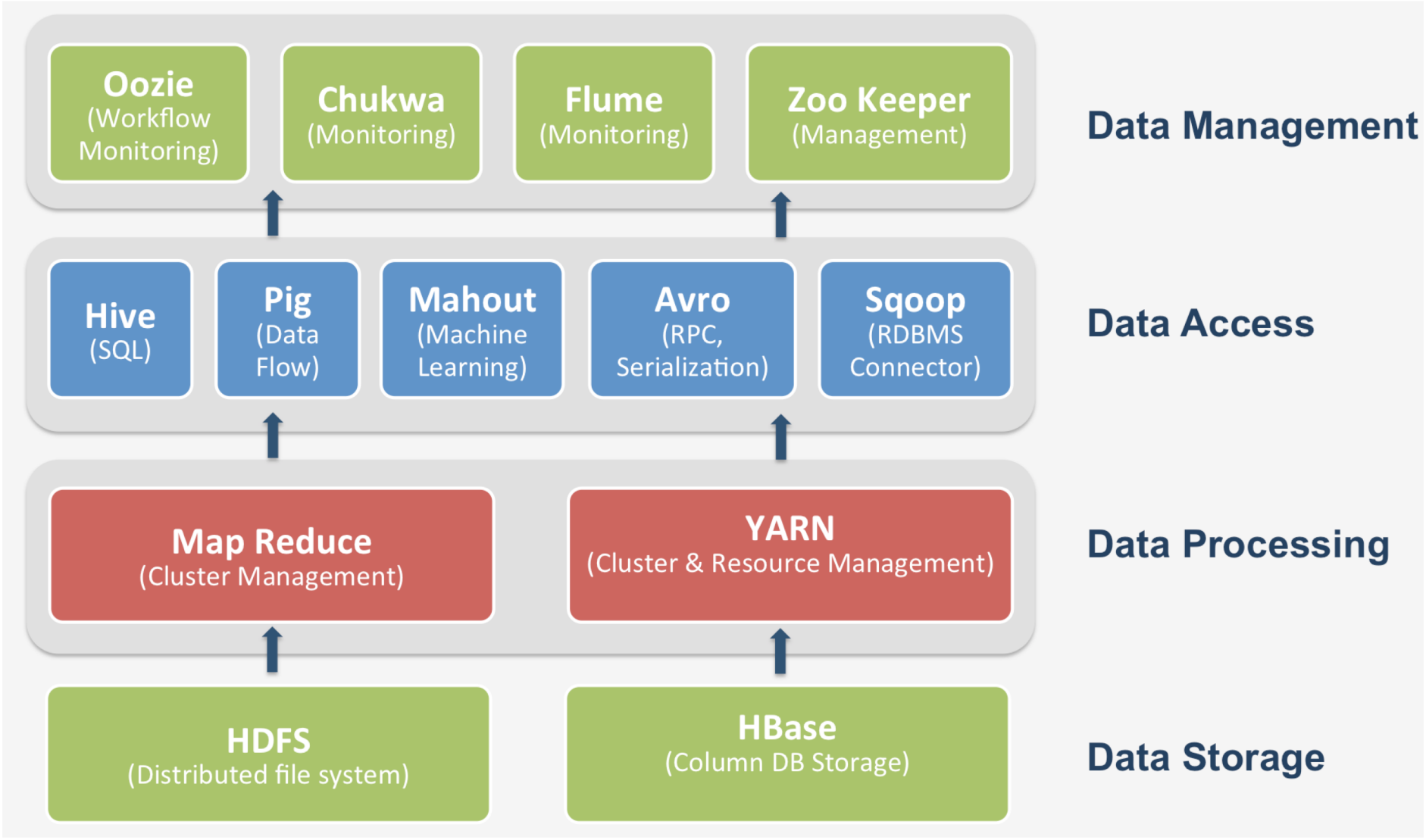
Big Data with Hadoop Architecture

Hadoop Cluster



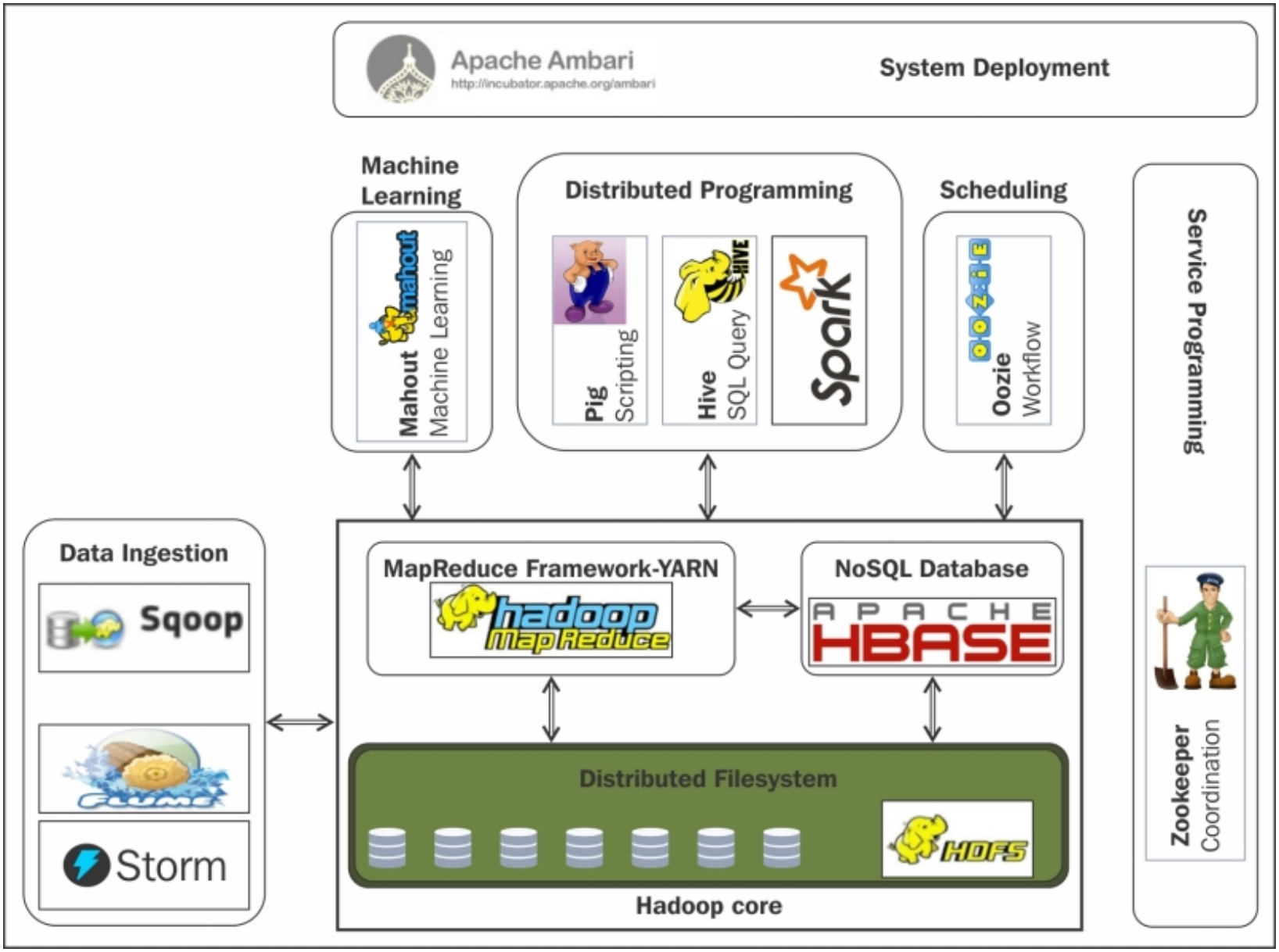
Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Hadoop Ecosystem



Source: https://savvycomsoftware.com/what-you-need-to-know-about-hadoop-and-its-ecosystem/

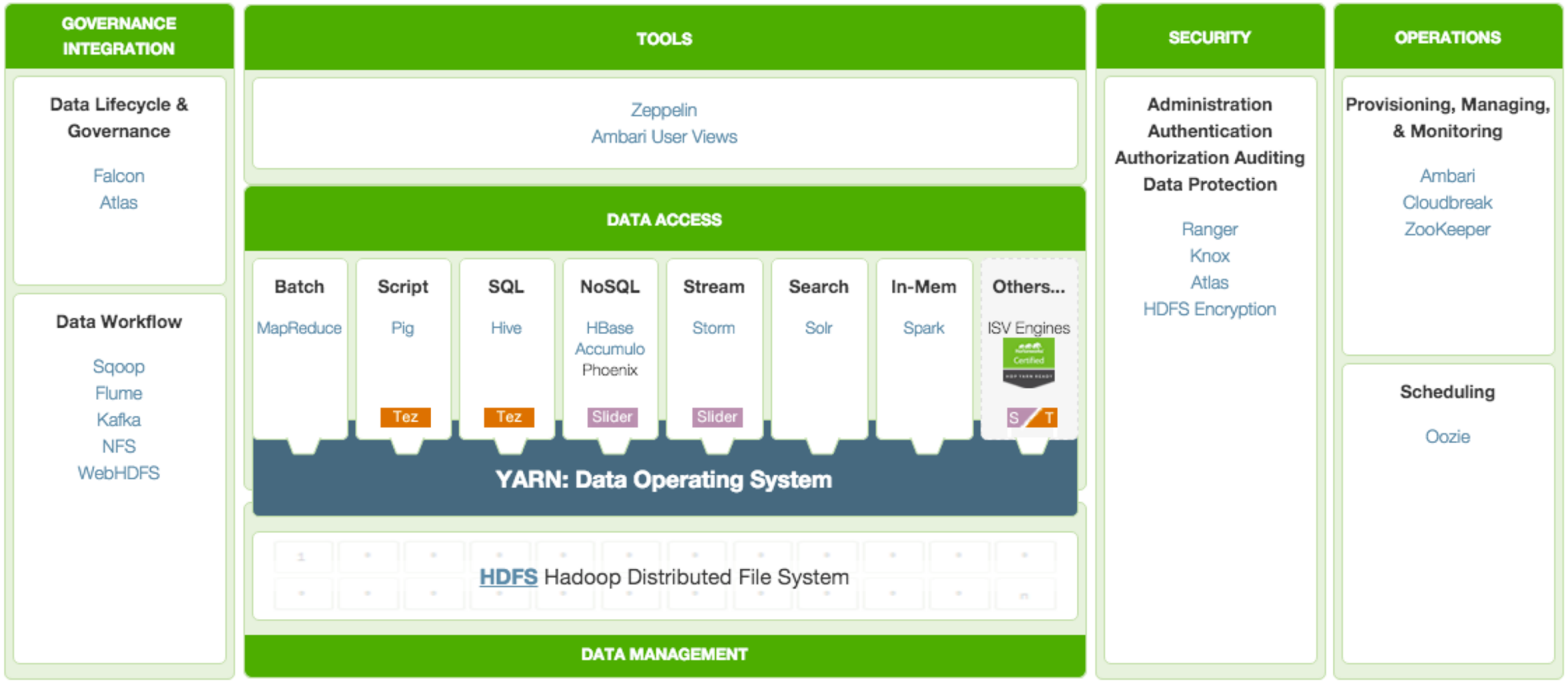
Hadoop Ecosystem



Source: Shiva Achari (2015), Hadoop Essentials - Tackling the Challenges of Big Data with Hadoop, Packt Publishing

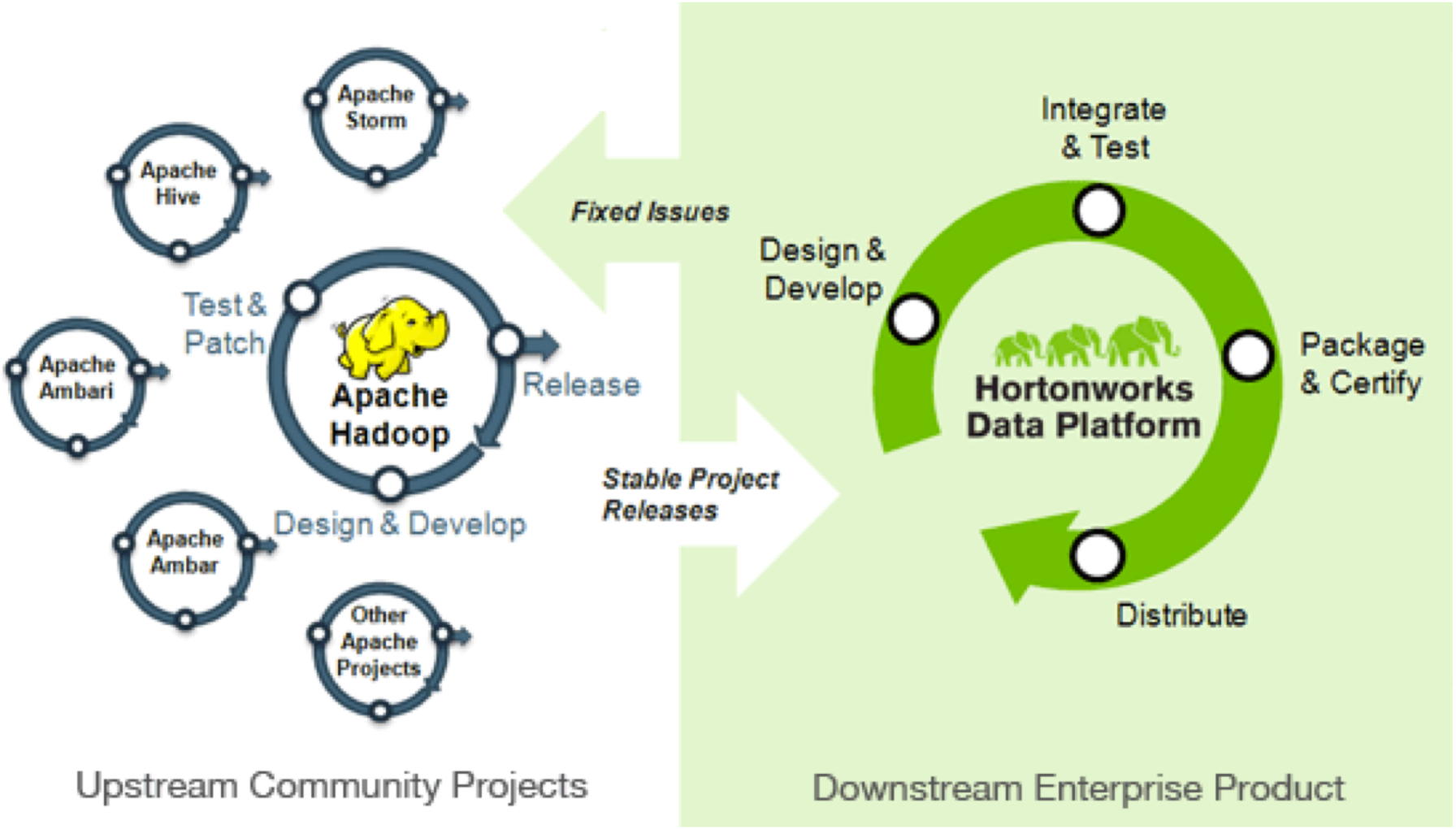
HDP (Hortonworks Data Platform)

A Complete Enterprise Hadoop Data Platform



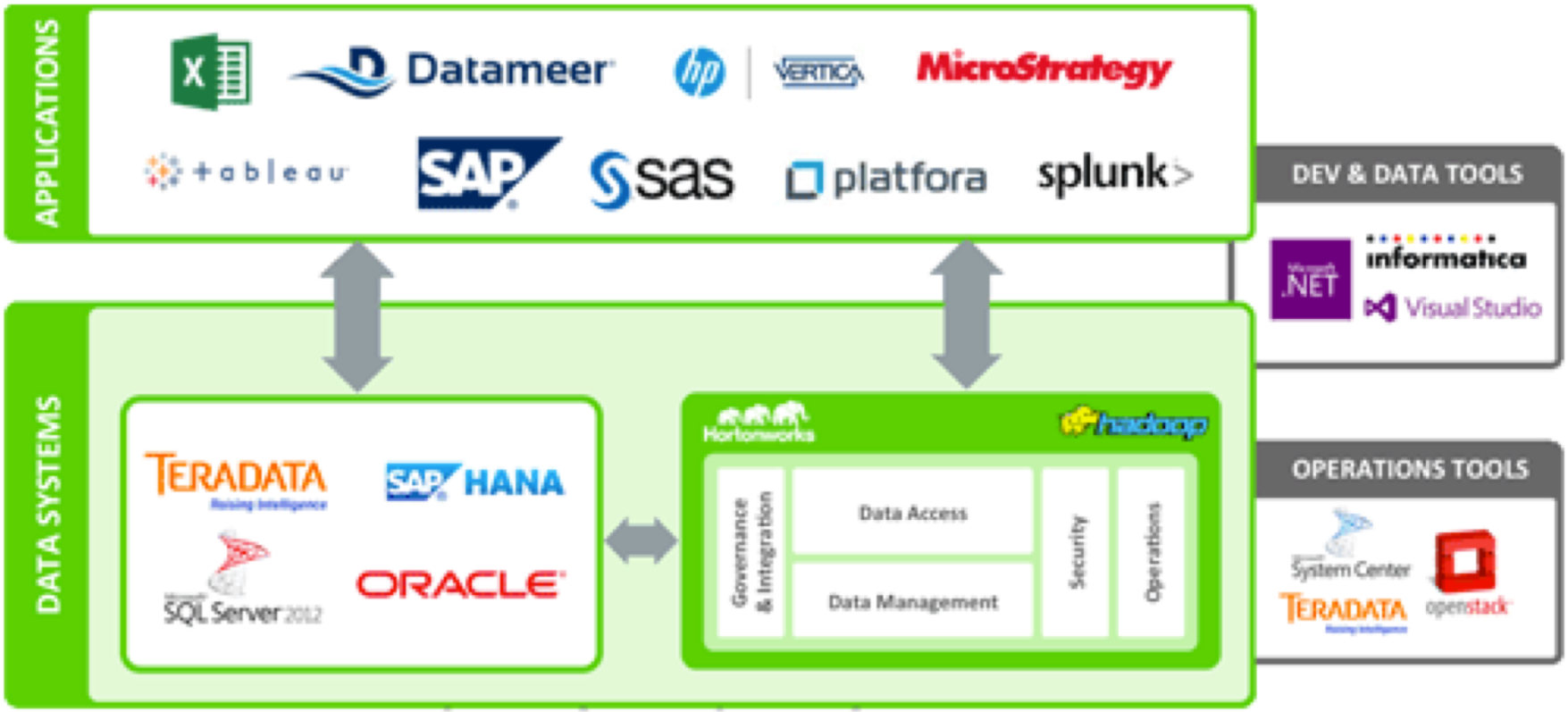
Source: http://hortonworks.com/hdp/

Apache Hadoop
Hortonworks Data Platform



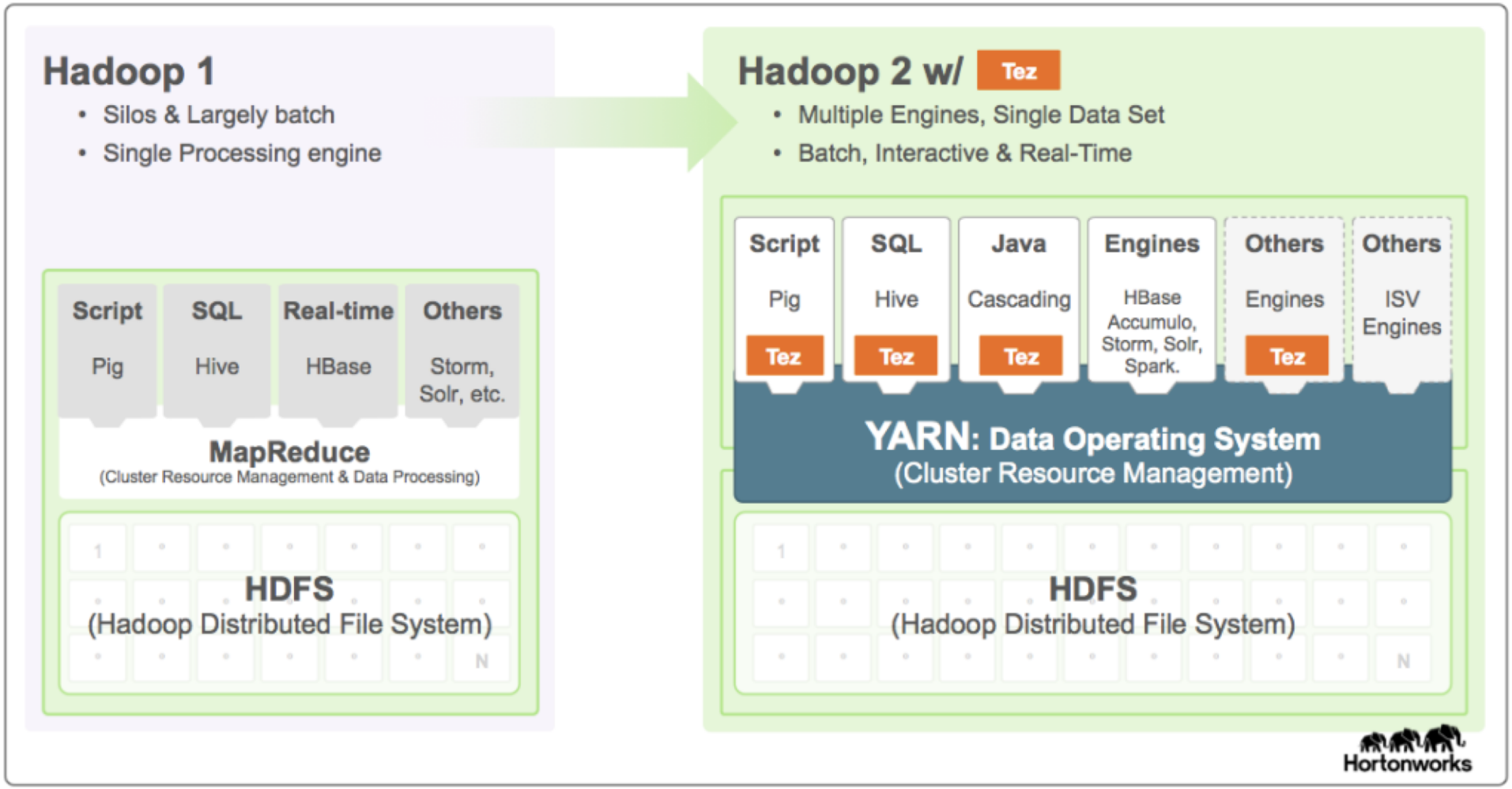
Source: http://hortonworks.com/hdp/

Hadoop and Data Analytics Tools



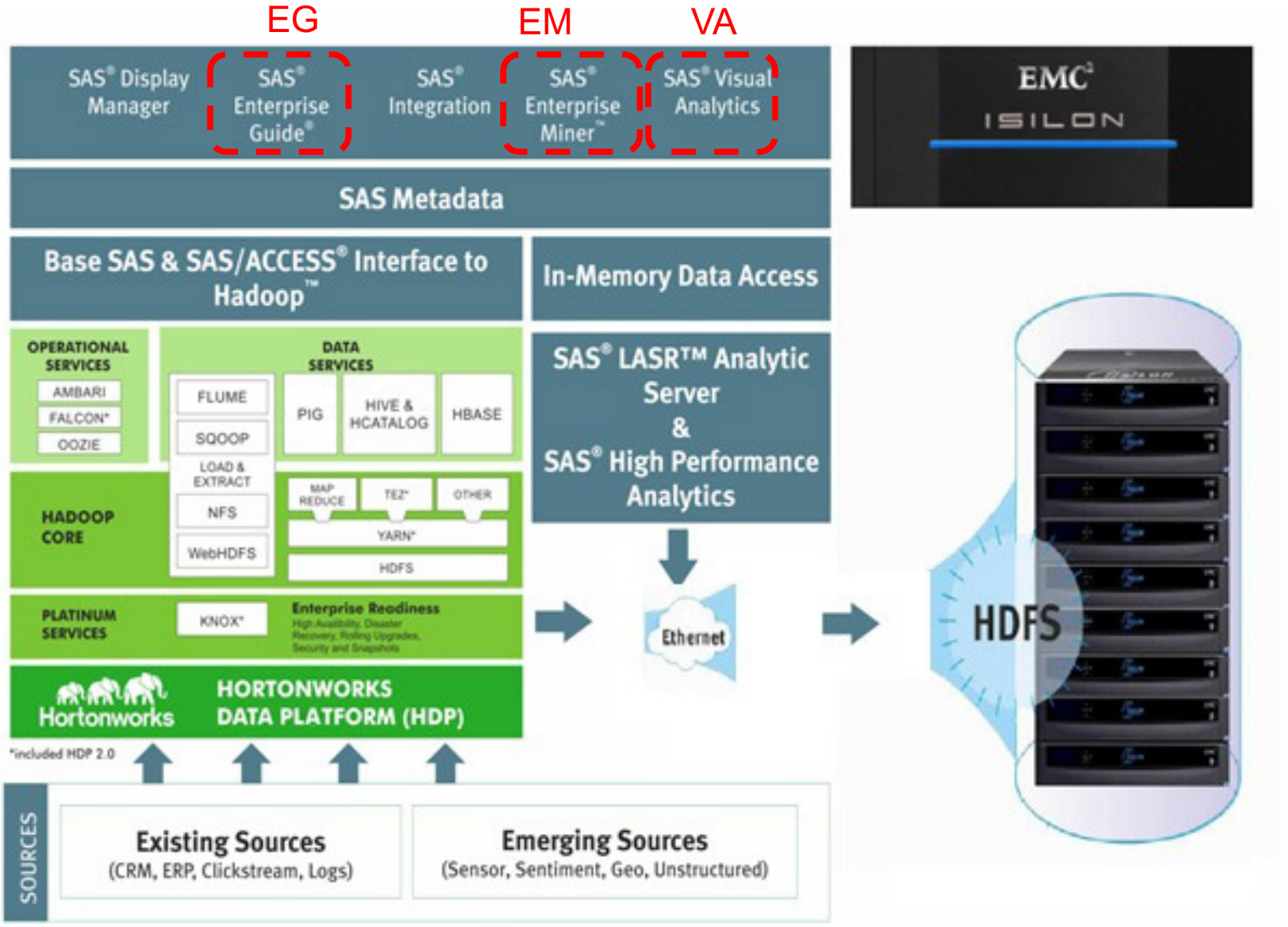
Source: http://hortonworks.com/hdp/

Hadoop 1 → Hadoop 2



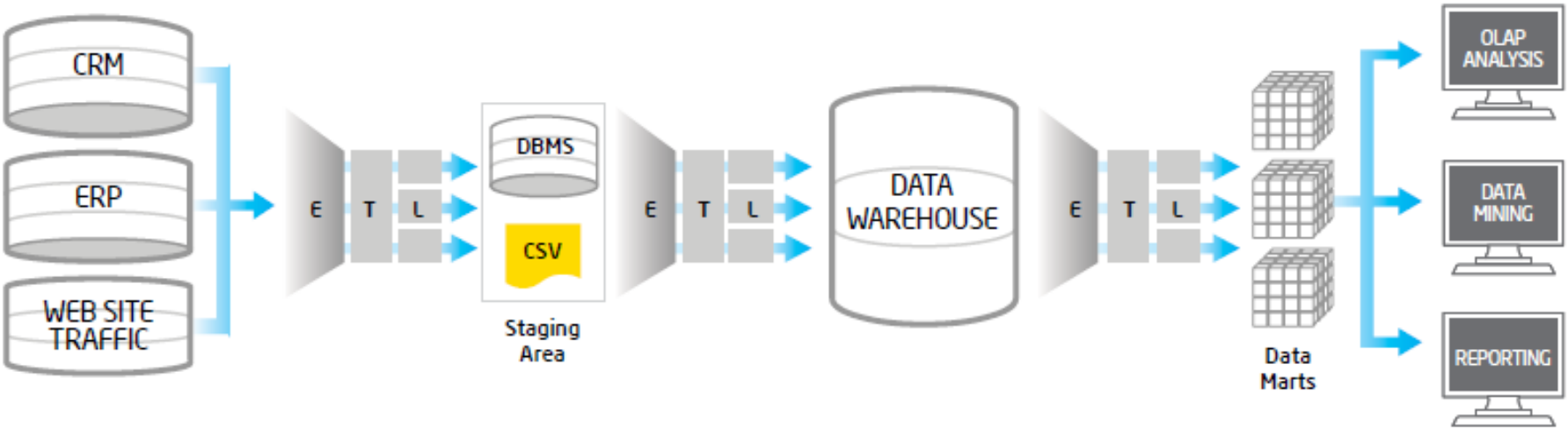
Source: http://hortonworks.com/hadoop/tez/

Big Data Solution



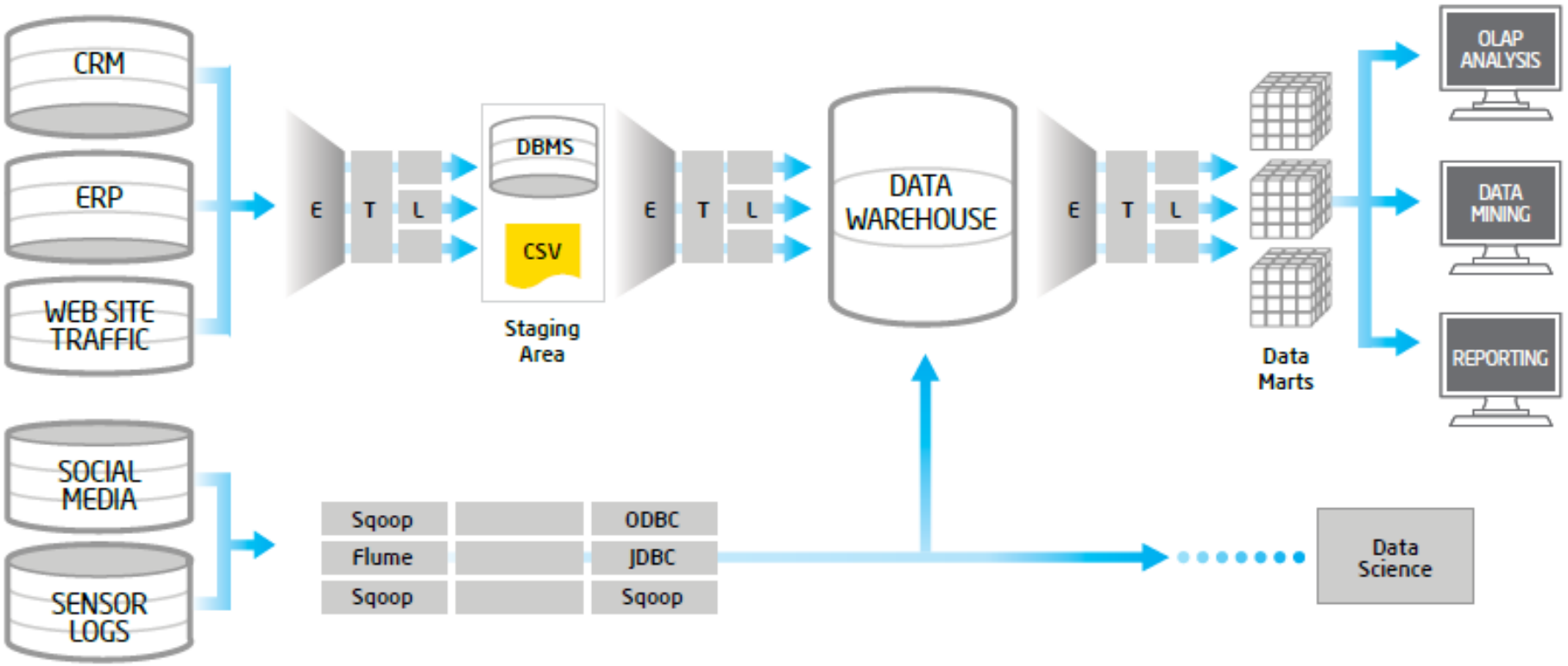
Source: http://www.newera-technologies.com/big-data-solution.html

Traditional ETL Architecture



Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Offload ETL with Hadoop (Big Data Architecture)



Source: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

Spark Ecosystem

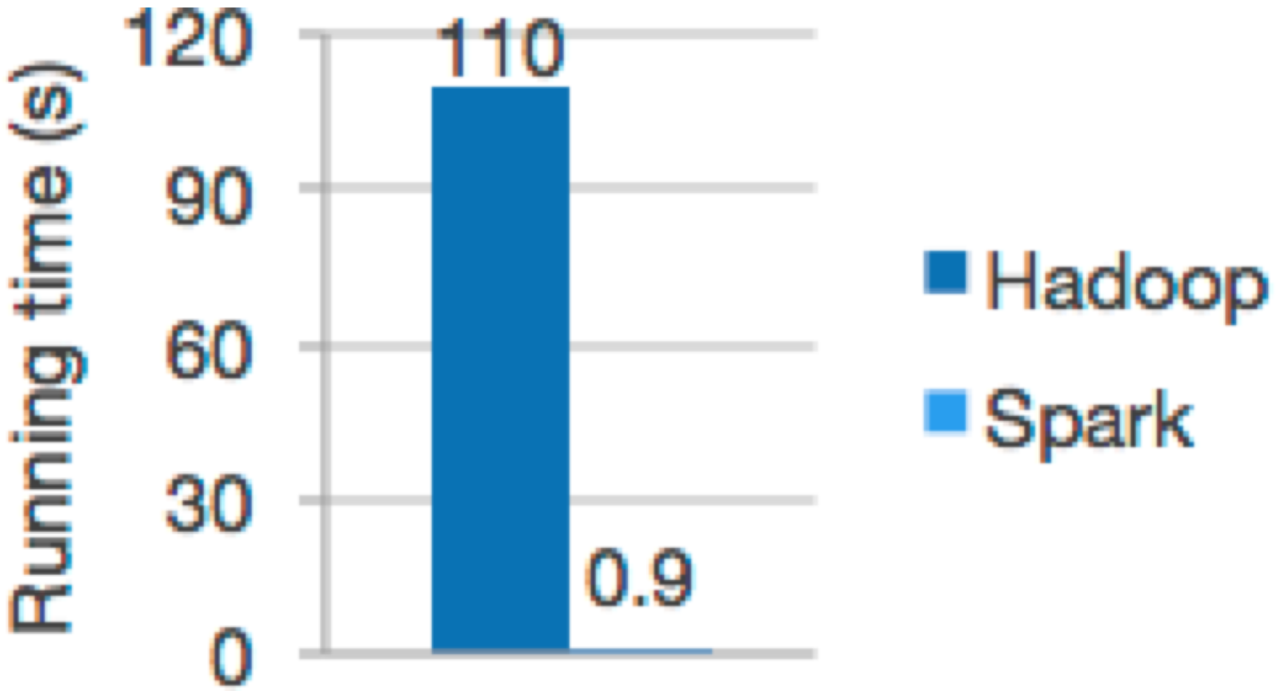
Lightning-fast cluster computing

Apache Spark

is a fast and general engine for large-scale data processing.

Source: http://spark.apache.org/

Logistic regression in Hadoop and Spark



Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

Source: http://spark.apache.org/

Ease of Use

- Write applications quickly in Java, Scala, Python, R.

Source: http://spark.apache.org/

Word count in Spark's Python API

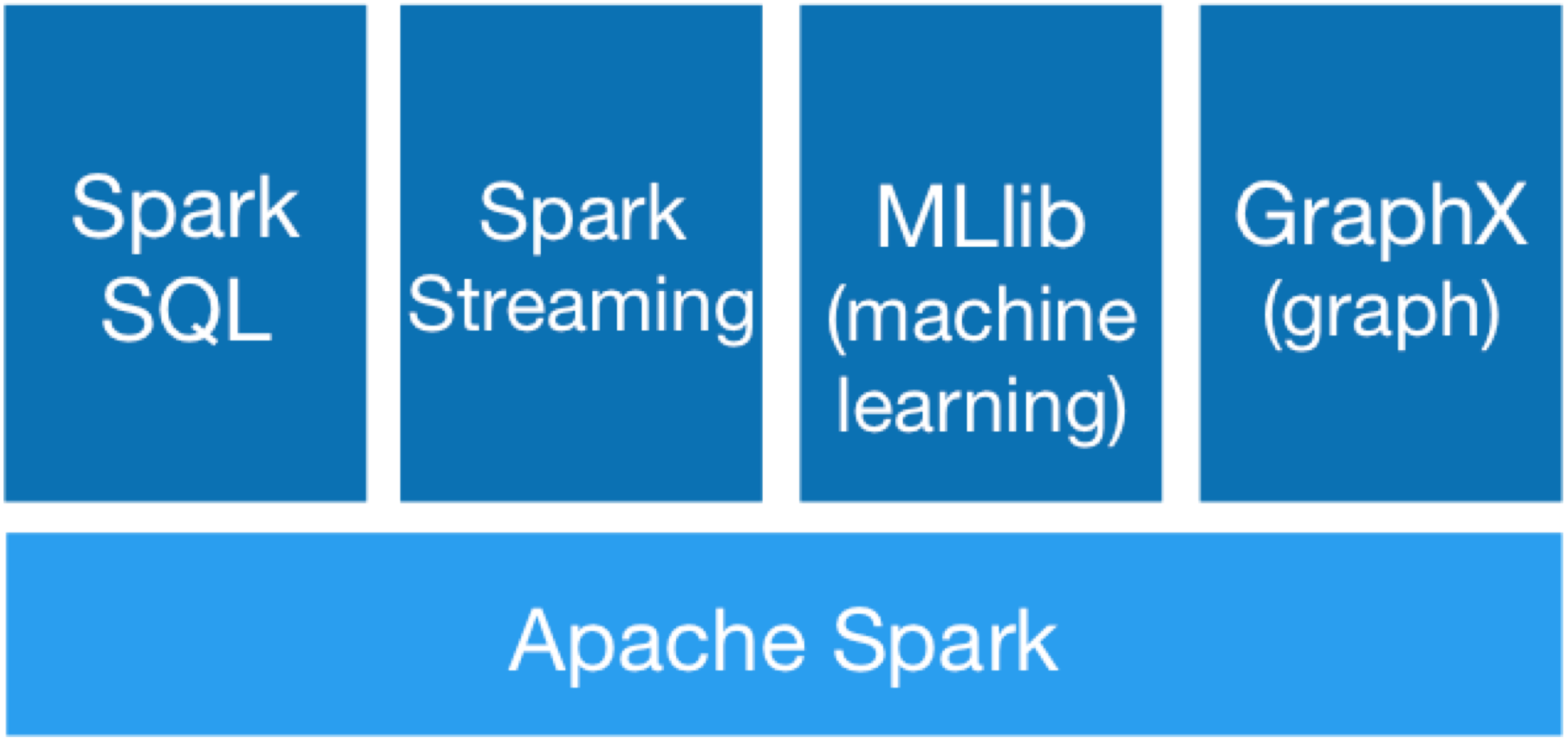
```
text_file = spark.textFile("hdfs://...")

text_file.flatMap(lambda line: line.split())
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a+b)
```

Source: http://spark.apache.org/

Spark and Hadoop

Source: http://spark.apache.org/

Spark Ecosystem



Source: http://spark.apache.org/

Spark Ecosystem

Spark SQL + DataFrames

Streaming

MLlib
Machine Learning

GraphX
Graph Computation

Spark Core API

R

SQL

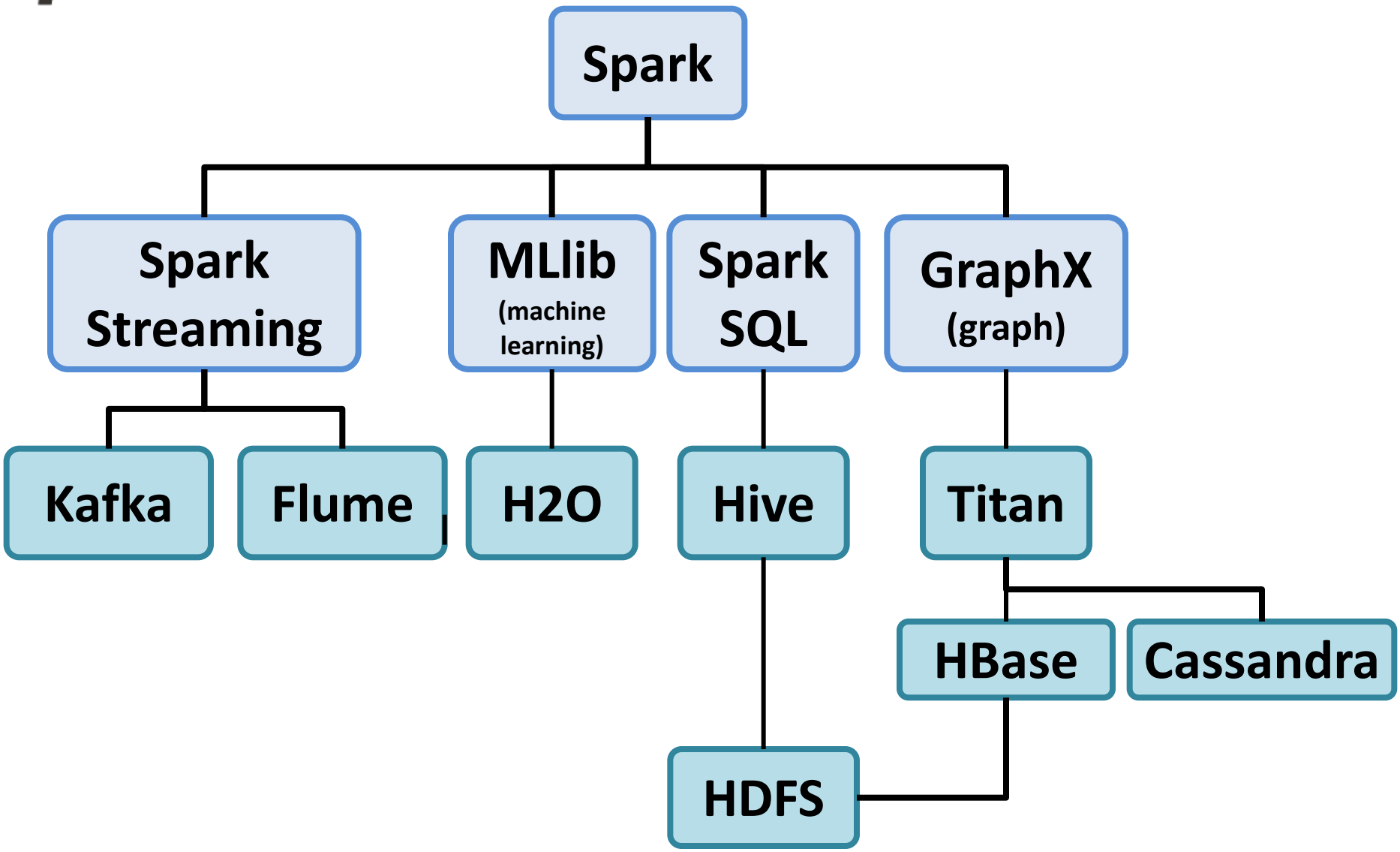
Python

Scala

Java

Source: https://databricks.com/spark/about

Spark Ecosystem



Source: Mike Frampton (2015), Mastering Apache Spark, Packt Publishing

SMACK Stack

- **Spark**
  - fast and general engine for distributed, large-scale data processing

- **Mesos**
  - cluster resource management system that provides efficient resource isolation and sharing across distributed applications

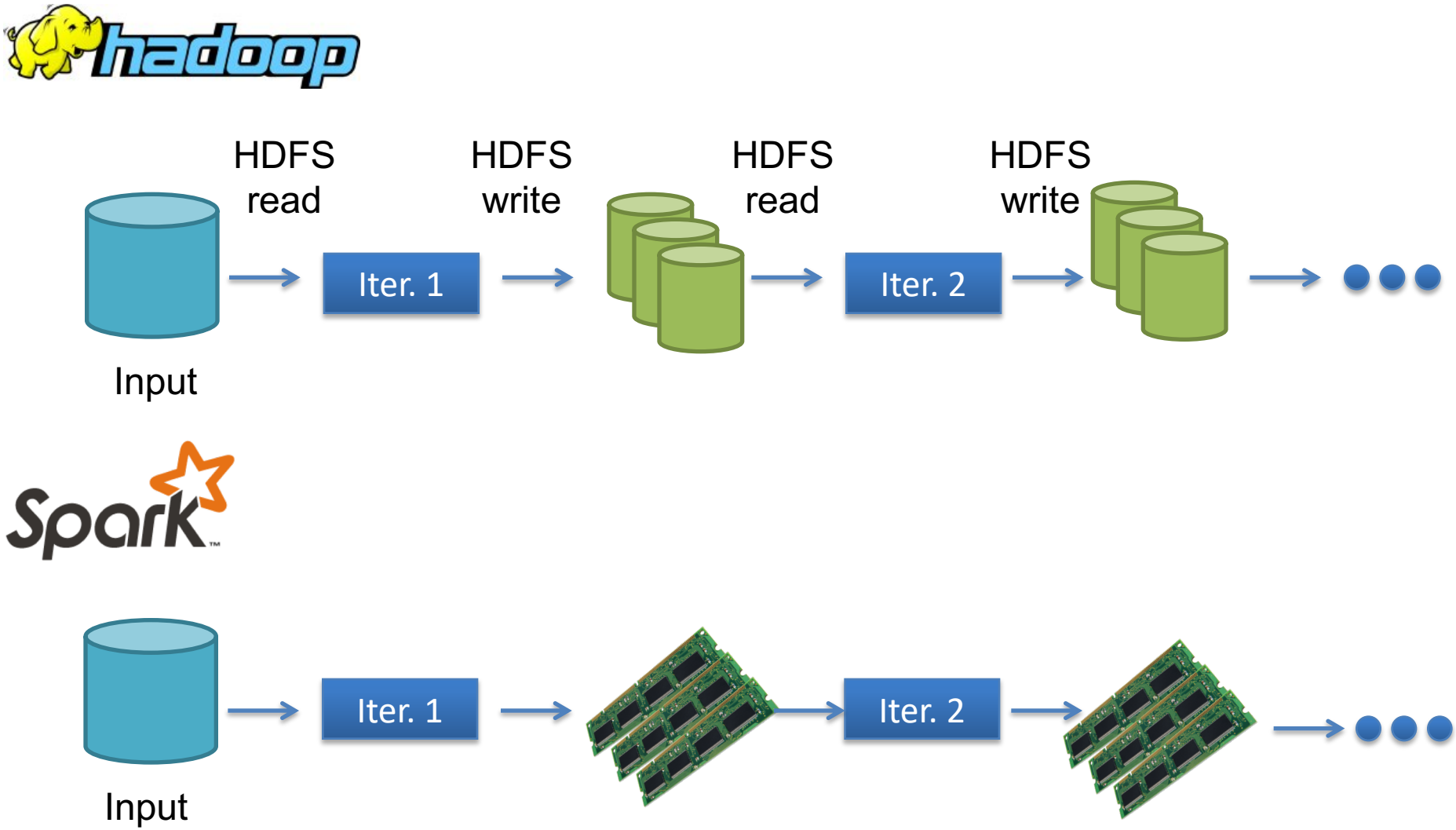
- **Akka**
  - a toolkit and runtime for building highly concurrent, distributed, and resilient message-driven applications on the JVM

- **Cassandra**
  - distributed, highly available database designed to handle large amounts of data across multiple datacenters

- **Kafka**
  - a high-throughput, low-latency distributed messaging system designed for handling real-time data feeds

Source: Anton Kirillov (2015), Data processing platforms architectures with Spark, Mesos, Akka, Cassandra and Kafka, Big Data AW Meetup

Hadoop vs. Spark



Source: Shiva Achari (2015), Hadoop Essentials - Tackling the Challenges of Big Data with Hadoop, Packt Publishing

Hadoop Distribution

- Apache Hadoop
  - http://hadoop.apache.org/
- Amazon Elastic Map Reduce (EMR)
  - https://aws.amazon.com/emr/
- Cloudera CDH
  - https://www.cloudera.com/downloads.html
- Hortonworks Sandbox
  - https://hortonworks.com/products/sandbox/

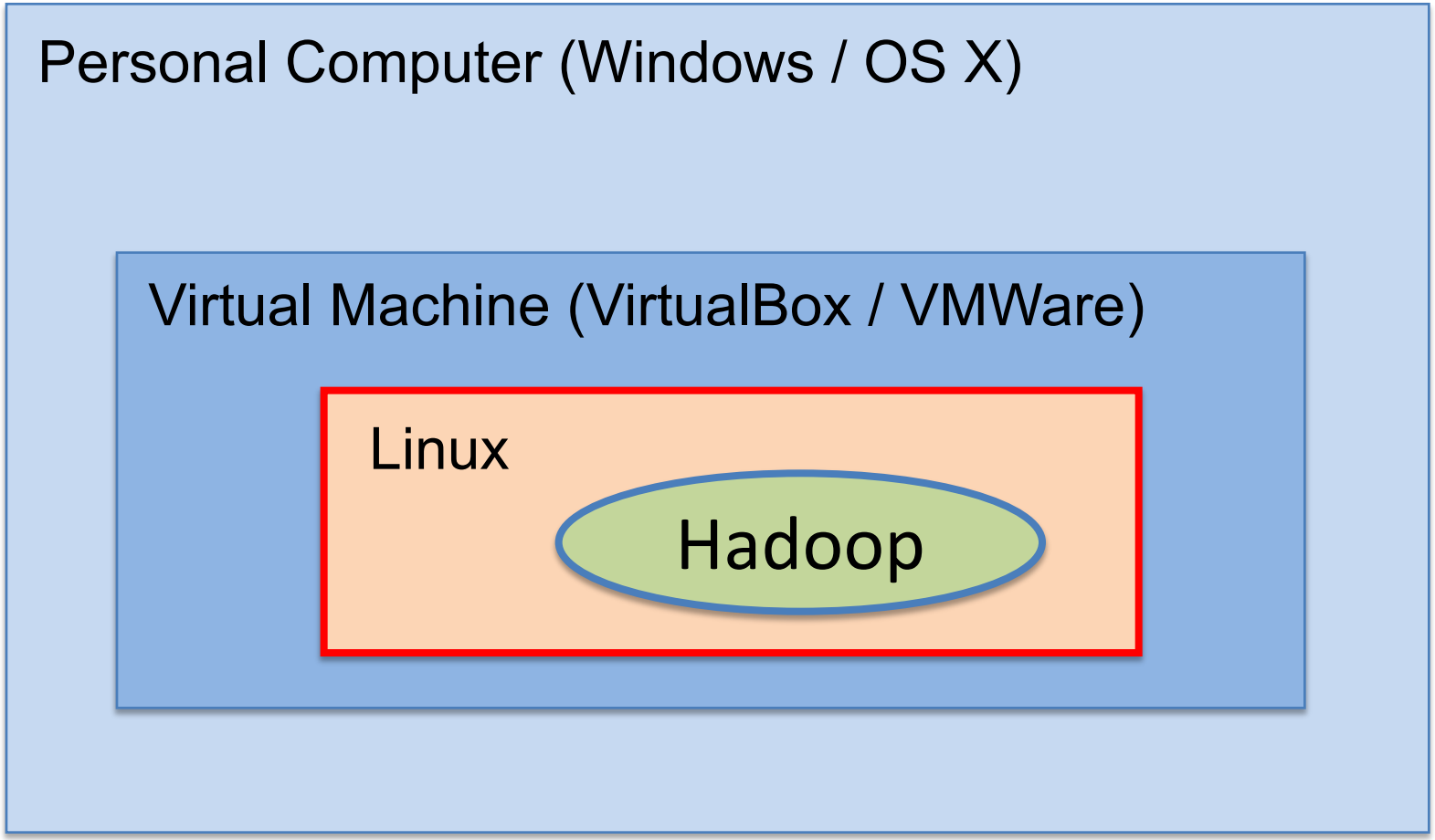
Steps to Install Hadoop on a Personal Computer (Windows/OS X)

Source: https://www.youtube.com/watch?v=rO-V1mxhzcM&list=PLyZEf-TOnZen8E5m5TIpIsdok2fyKDNRa&index=5

Hodoop: Linux Based Software

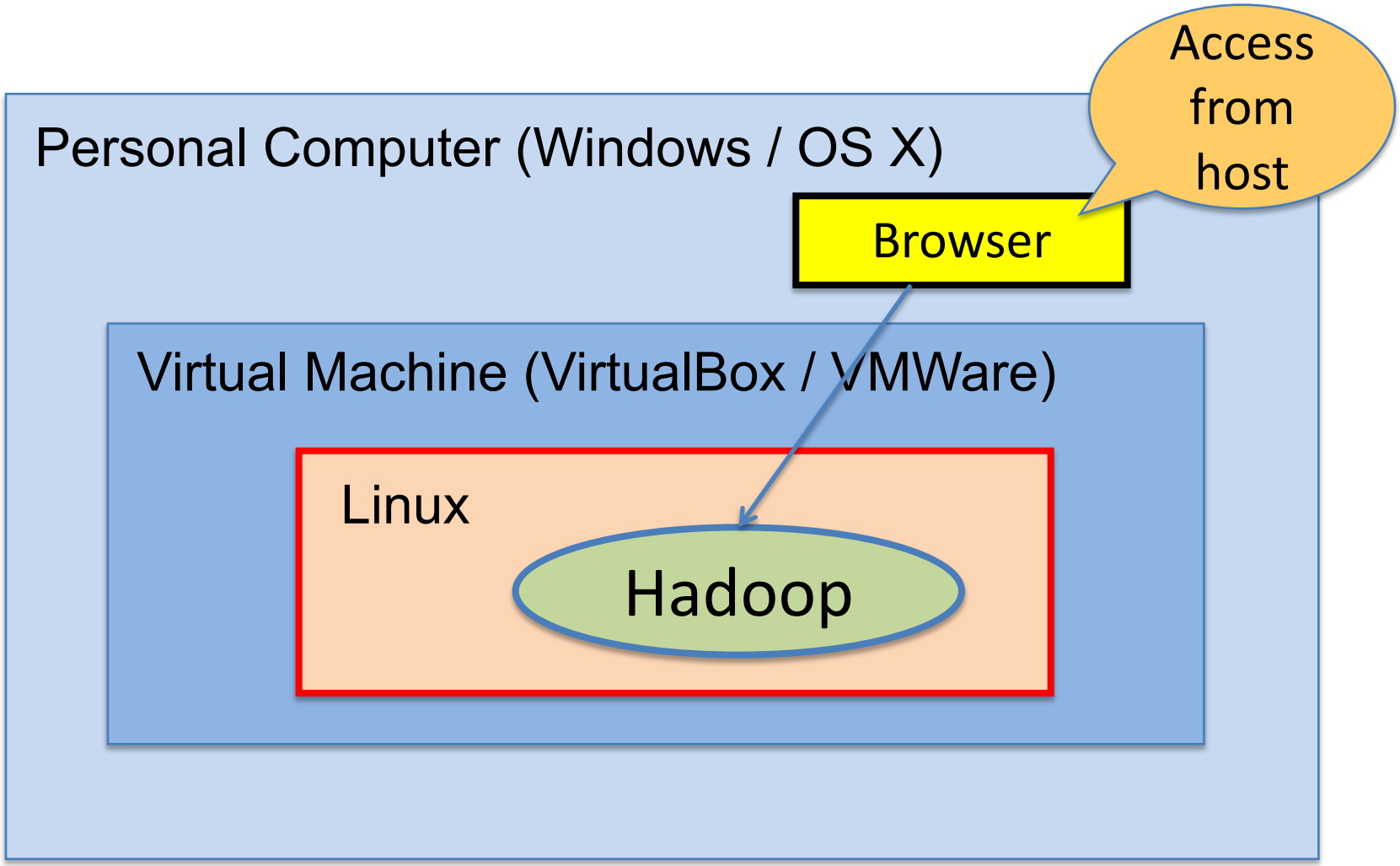
Source: https://www.youtube.com/watch?v=rO-V1mxhzcM&list=PLyZEf-TOnZen8E5m5TIpIsdok2fyKDNRa&index=5

Appliance



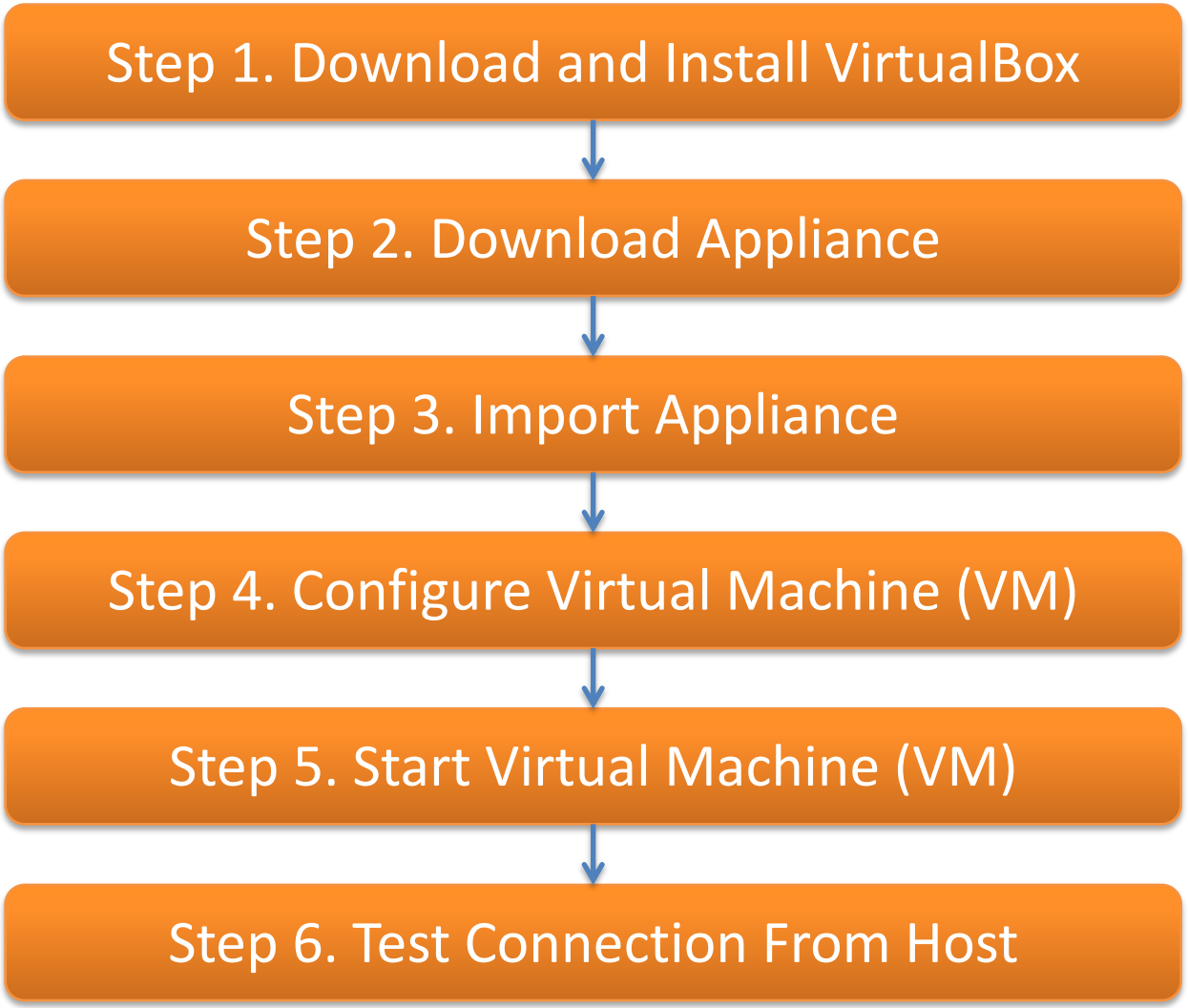
Source: https://www.youtube.com/watch?v=rO-V1mxhzcM&list=PLyZEf-TOnZen8E5m5TIpIsdok2fyKDNRa&index=5

Connection to Hadoop



Source: https://www.youtube.com/watch?v=rO-V1mxhzcM&list=PLyZEf-TOnZen8E5m5TIpIsdok2fyKDNRa&index=5

Steps to Install Hadoop on a Personal Computer (Windows/OS X)



Source: https://www.youtube.com/watch?v=rO-V1mxhzcM&list=PLyZEf-TOnZen8E5m5TIpIsdok2fyKDNRa&index=5

Virtual Box

https://www.virtualbox.org/

Steps to Install Hadoop on a Personal Computer (Windows/OS X)

Step 1. Download and Install VirtualBox

Step 2. Download Appliance

Hortonworks Sandbox

Step 3. Import Appliance

Step 4. Configure Virtual Machine (VM)

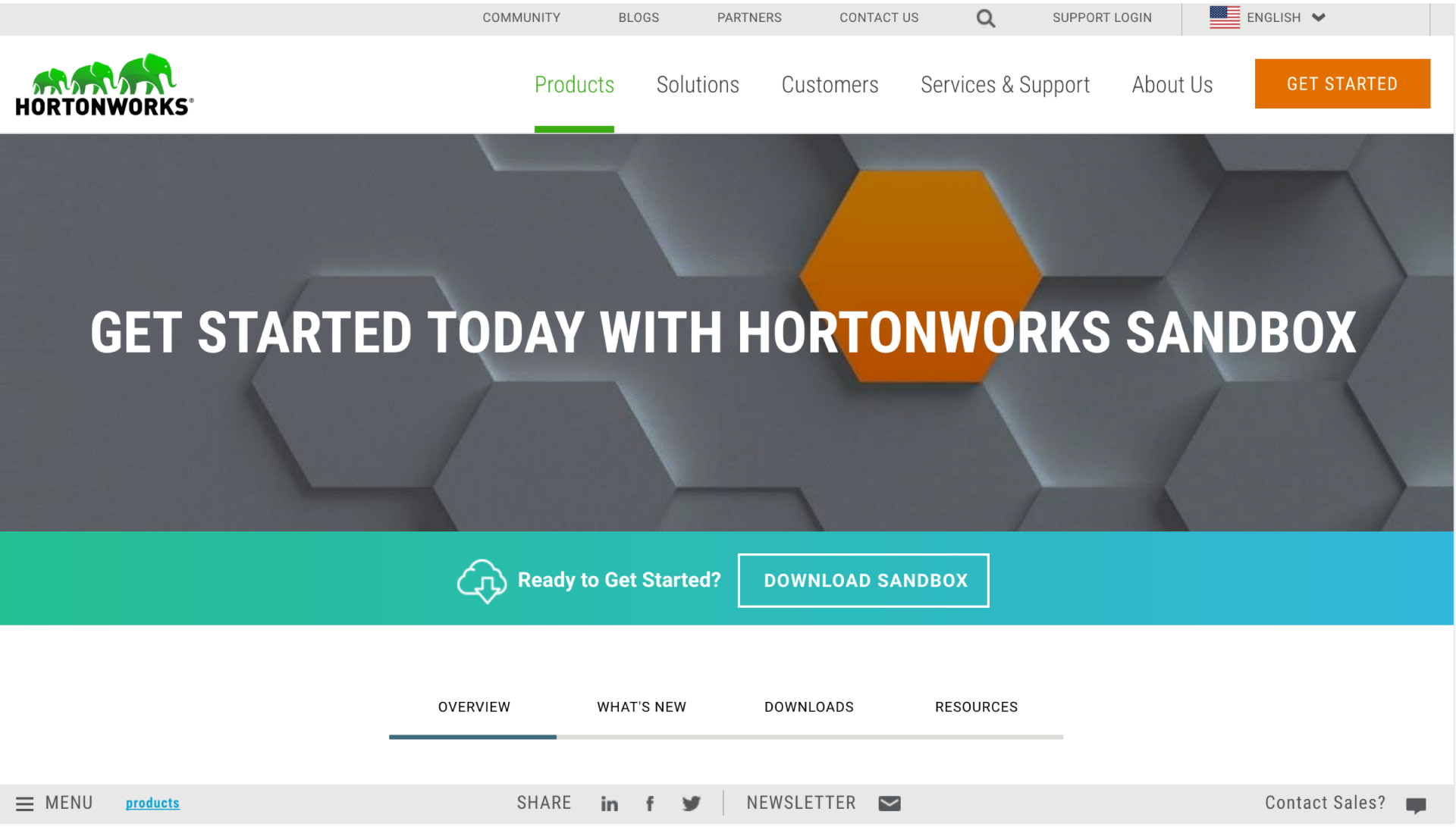
Step 5. Start Virtual Machine (VM)

Step 6. Test Connection From Host

Source: https://www.youtube.com/watch?v=rO-V1mxhzcM&list=PLyZEf-TOnZen8E5m5TIpIsdok2fyKDNRa&index=5

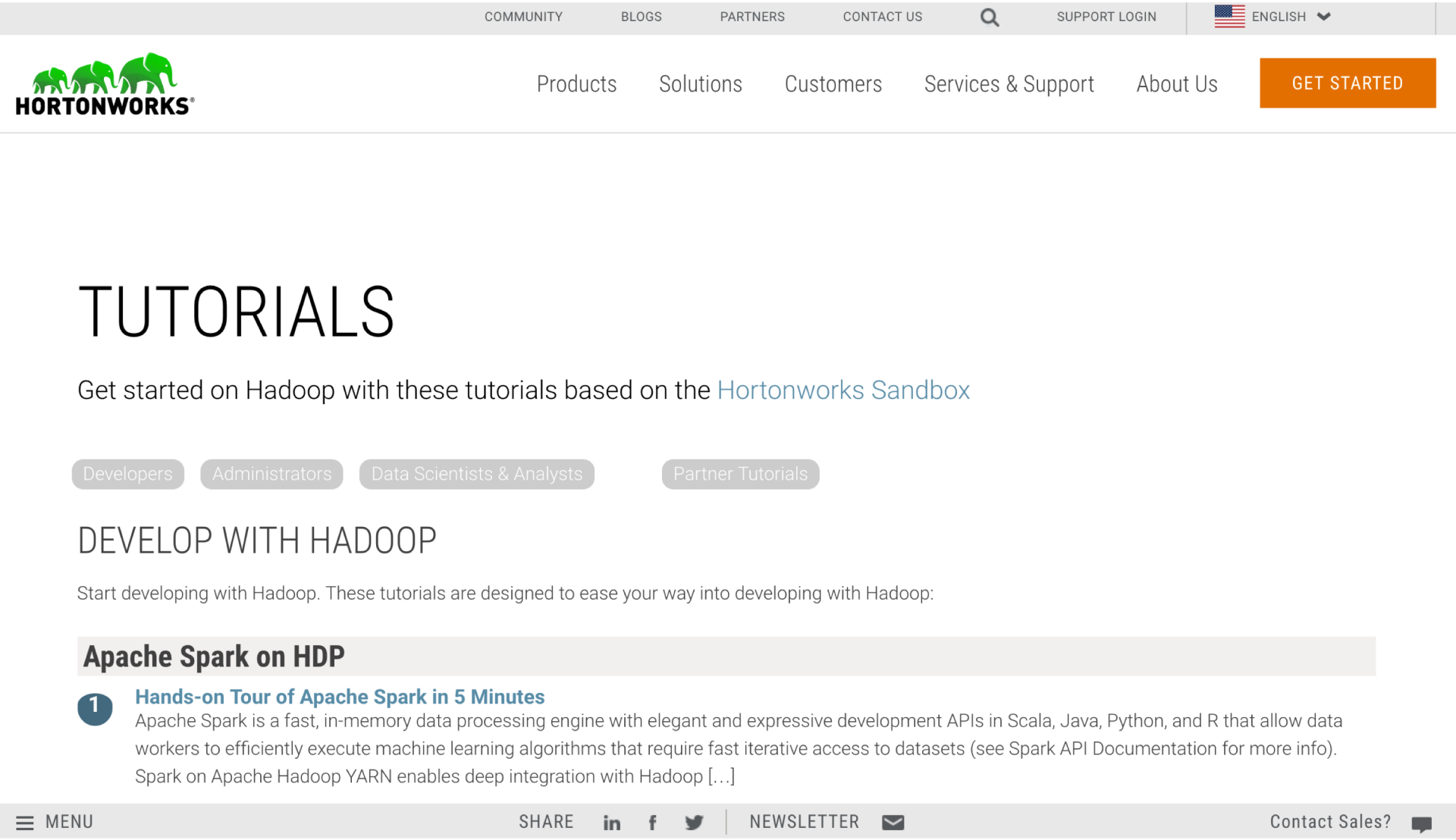
Hortonworks Sandbox

The easiest way to get started with Enterprise Hadoop



http://hortonworks.com/products/hortonworks-sandbox/#install

Get started on Hadoop with these tutorials based on the Hortonworks Sandbox



http://hortonworks.com/tutorials/

Apache Hadoop

Welcome to Apache™ Hadoop®!

What Is Apache Hadoop?

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

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.

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.

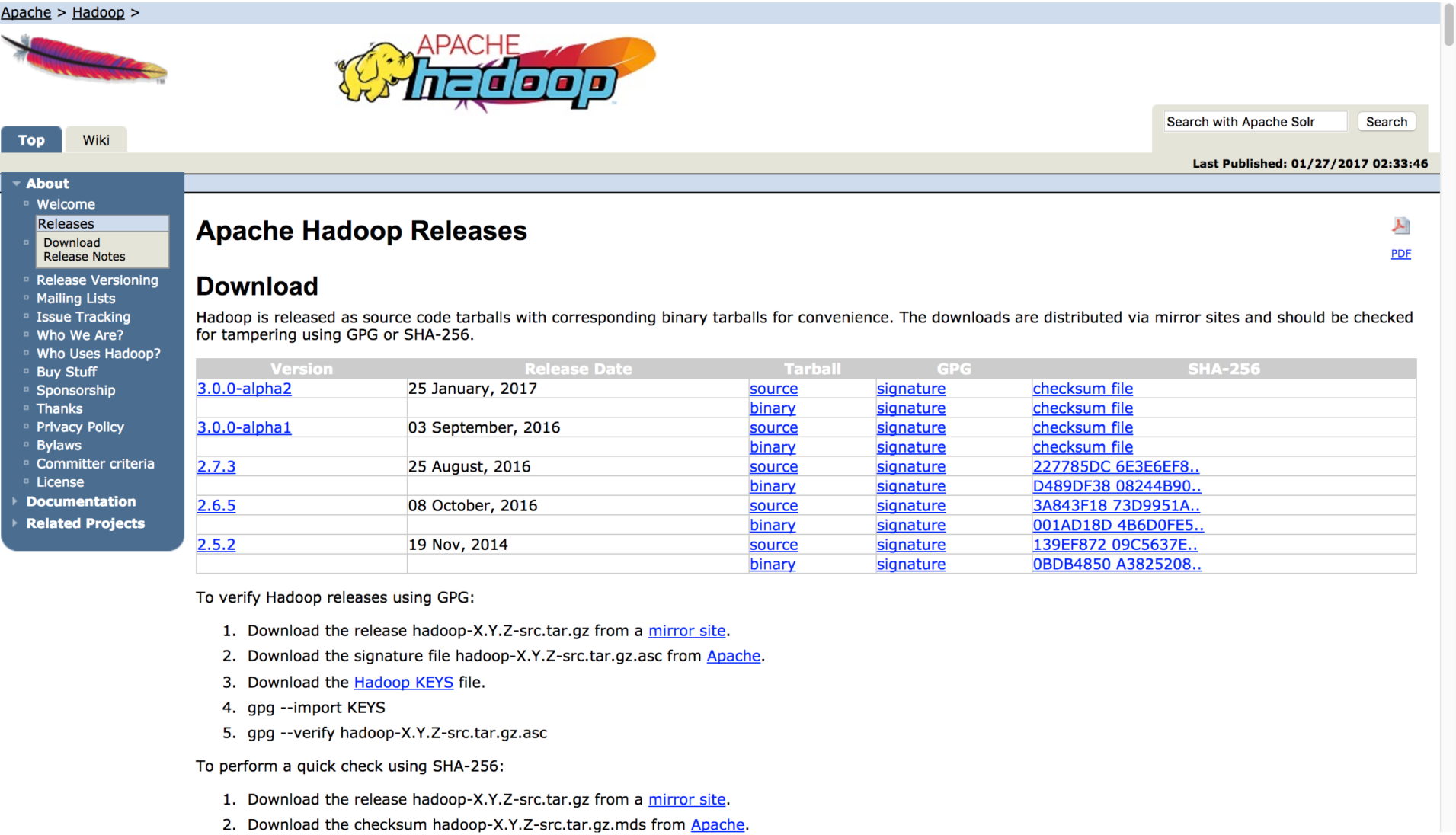
Other Hadoop-related projects at Apache include:

- **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.

http://hadoop.apache.org/

Apache Hadoop

http://hadoop.apache.org/releases.html#Download



Apache > Hadoop >

Top | Wiki

Search with Apache Solr | Search

Last Published: 01/27/2017 02:33:46

- About
  - Welcome
  - Releases
    - Download
    - Release Notes
  - Release Versioning
  - Mailing Lists
  - Issue Tracking
  - Who We Are?
  - Who Uses Hadoop?
  - Buy Stuff
  - Sponsorship
  - Thanks
  - Privacy Policy
  - Bylaws
  - Committer criteria
  - License
- Documentation
- Related Projects

Apache Hadoop Releases

PDF

Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-256.

Version	Release Date	Tarball	GPG	SHA-256
3.0.0-alpha2	25 January, 2017	source	signature	checksum file
		binary	signature	checksum file
3.0.0-alpha1	03 September, 2016	source	signature	checksum file
		binary	signature	checksum file
2.7.3	25 August, 2016	source	signature	227785DC 6E3E6EF8..
		binary	signature	D489DF38 08244B90..
2.6.5	08 October, 2016	source	signature	3A843F18 73D9951A..
		binary	signature	001AD18D 4B6D0FE5..
2.5.2	19 Nov, 2014	source	signature	139EF872 09C5637E..
		binary	signature	0BDB4850 A3825208..

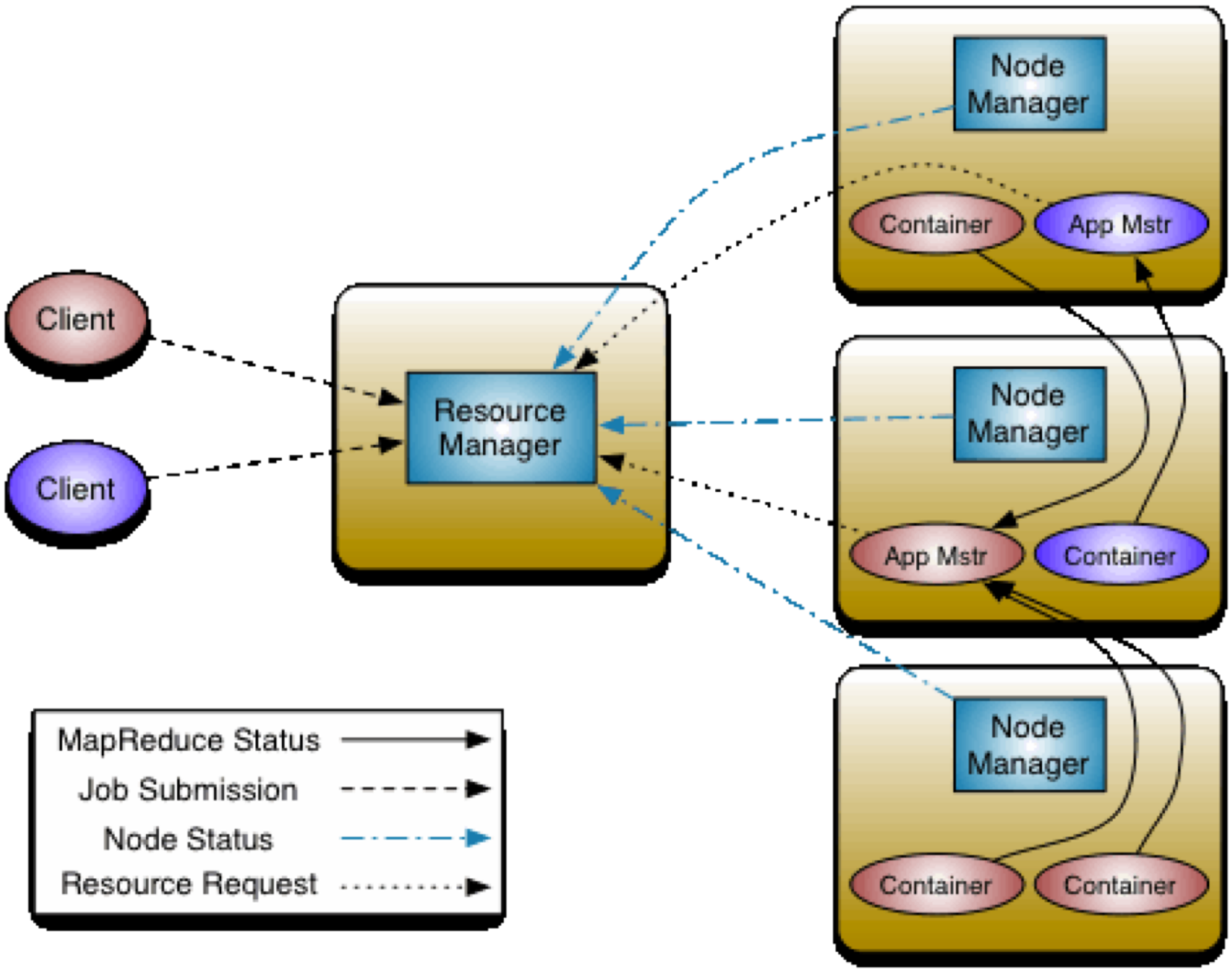
To verify Hadoop releases using GPG:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the signature file hadoop-X.Y.Z-src.tar.gz.asc from Apache.
3. Download the Hadoop KEYS file.
4. gpg --import KEYS
5. gpg --verify hadoop-X.Y.Z-src.tar.gz.asc

To perform a quick check using SHA-256:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the checksum hadoop-X.Y.Z-src.tar.gz.mds from Apache.

Apache Hadoop YARN



Source: http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Apache Spark

Lightning-fast unified analytics engine

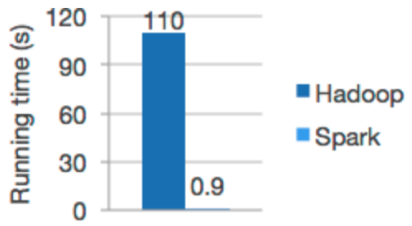
Download | Libraries | Documentation | Examples | Community | Developers | Apache Software Foundation

Apache Spark™ is a unified analytics engine for large-scale data processing.

Speed

Run workloads 100x faster.

Apache Spark achieves high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.



Logistic regression in Hadoop and Spark

Ease of Use

Write applications quickly in Java, Scala, Python, R, and SQL.

Spark offers over 80 high-level operators that make it easy to build parallel

```
df = spark.read.json("logs.json")
df.where("age > 21")
  .select("name.first").show()
```

Spark's Python DataFrame API
Read JSON files with automatic schema inference

Latest News

Spark 2.3.2 released (Sep 24, 2018)

Spark+AI Summit (October 2-4th, 2018, London) agenda posted (Jul 24, 2018)

Spark 2.2.2 released (Jul 02, 2018)

Spark 2.1.3 released (Jun 29, 2018)

Archive

APACHECON North America
September 24-27, 2018
Montréal, Canada

Download Spark

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

http://spark.apache.org/

databricks

databricks PRODUCT SPARK SOLUTIONS CUSTOMERS COMPANY BLOG RESOURCES Partners Training Sign Up

Apache Spark™ is a powerful open source processing engine built around speed, ease of use, and sophisticated analytics. It was originally developed at UC Berkeley in 2009.

The largest open source project in data processing.

Since its release, Spark has seen rapid adoption by enterprises across a wide range of industries. Internet powerhouses such as Yahoo, Baidu, and Tencent, have eagerly deployed Spark at massive scale, collectively processing multiple petabytes of data on clusters of over 8,000 nodes. It has quickly become the largest open source community in big data, with over 750 contributors from 200+ organizations.

The creators of Spark founded Databricks in 2013.

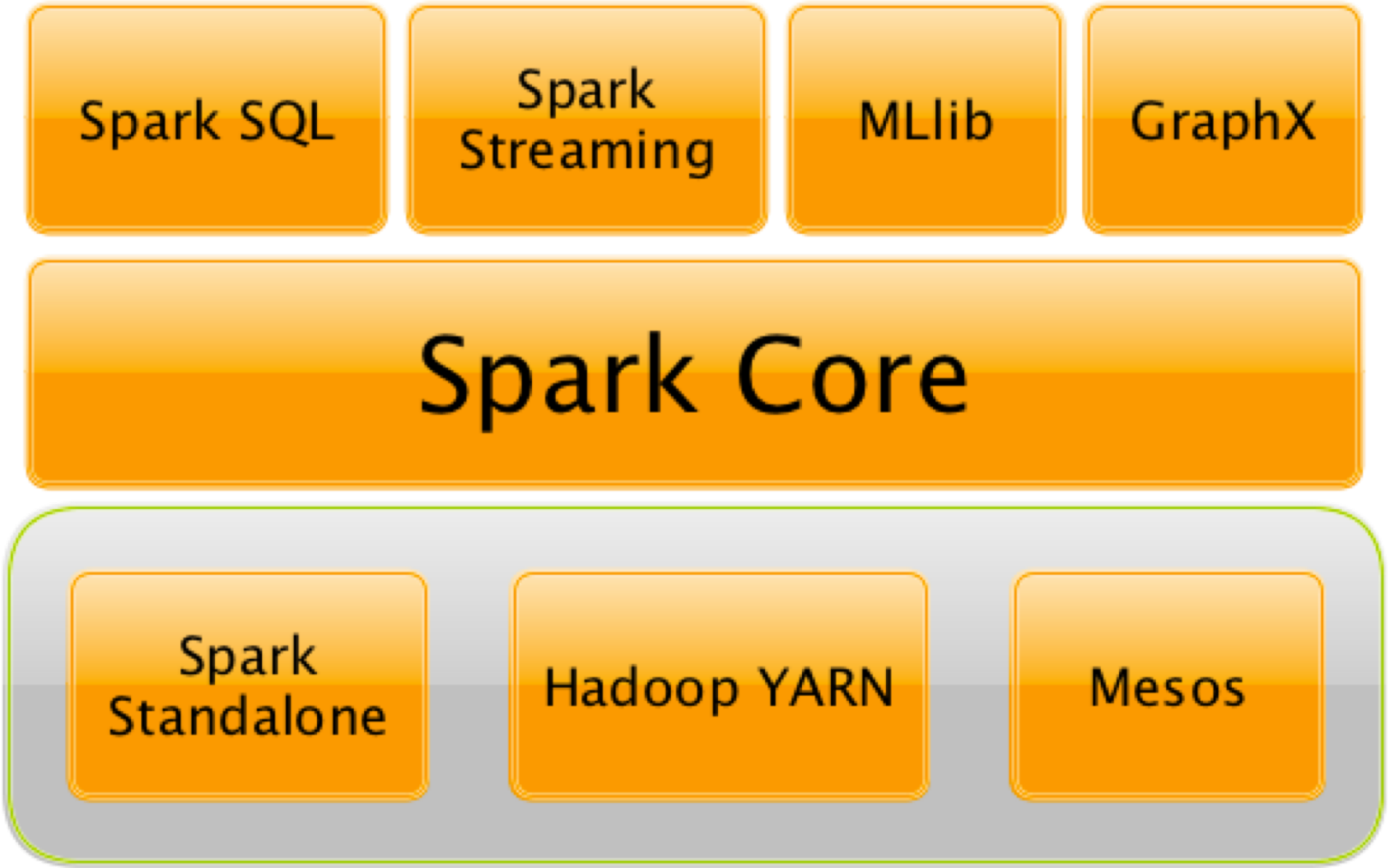
Databricks continues to grow the Spark project by contributing broadly, with both roadmap development and community evangelism.

"At Databricks, we're working hard to make Spark easier to use and run than ever, through our efforts on both the Spark codebase and support materials around it. All of our work on Spark is open source and goes directly to Apache."

— Matei Zaharia, VP, Apache Spark, Founder & CTO, Databricks

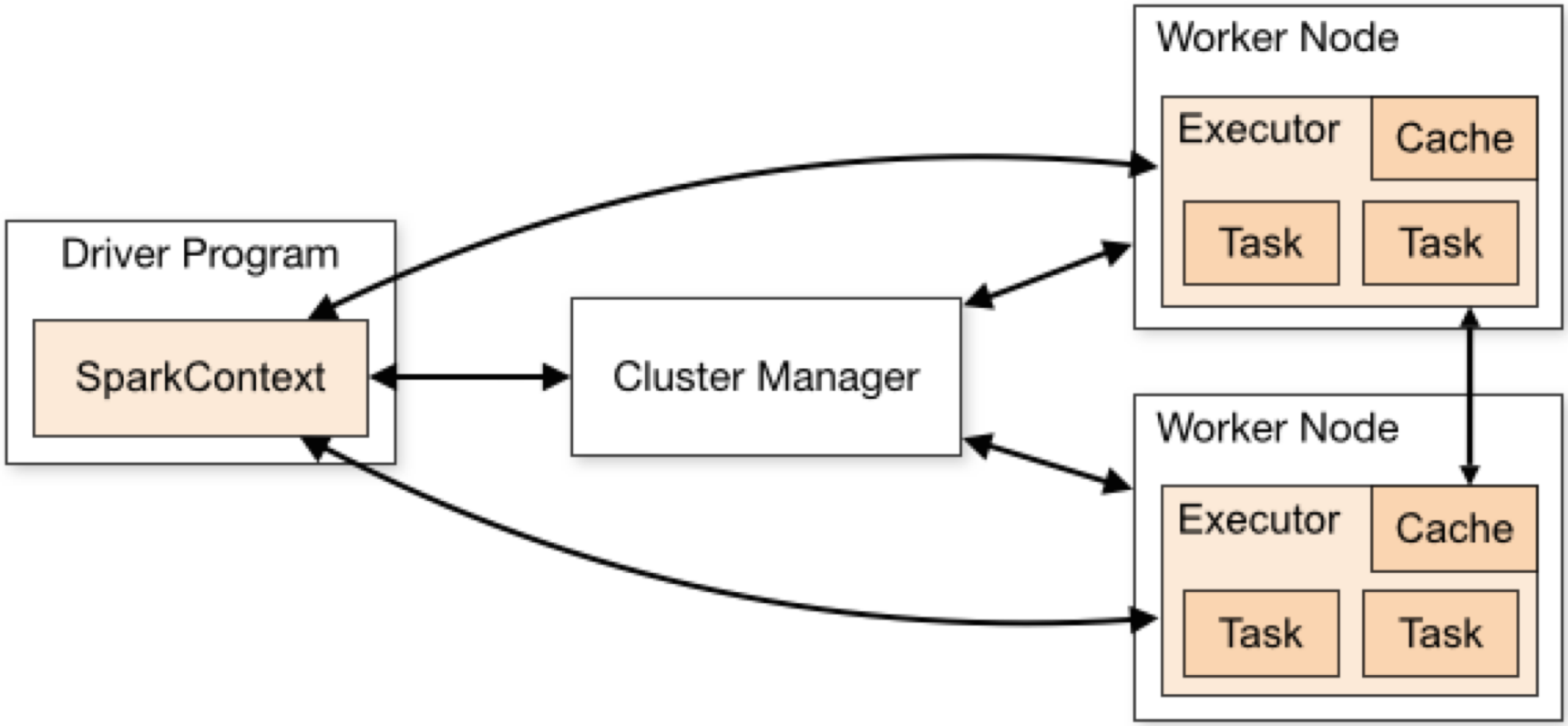
https://databricks.com/spark/about

The Spark Platform



Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-overview.html

Spark Cluster Overview



Source: http://spark.apache.org/docs/latest/cluster-overview.html

3 Types of Spark Cluster Manager

- **Standalone**
  - a simple cluster manager included with Spark that makes it easy to set up a cluster.
- **Apache Mesos**
  - a general cluster manager that can also run Hadoop MapReduce and service applications.
- **Hadoop YARN**
  - the resource manager in Hadoop 2.

Spark's EC2 launch scripts make it easy to launch a standalone cluster on Amazon EC2.

Source: http://spark.apache.org/docs/latest/cluster-overview.html

Running Spark on cluster
Cluster Managers (Schedulers)

- Spark's own Standalone cluster manager
- Hadoop YARN
- Apache Mesos

Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-cluster.html

Spark Developer Resources

Spark Developer Resources

Databricks provides a number of free resources online for Spark training, including course materials, video archives, sample apps, knowledge base, etc.

Note that course materials for these workshops are provided online under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.

Content Archives

Highlights of recent blogs, videos, and other community content:

- Apache Spark channel on YouTube (meetup livestream archives)
- Spark events worldwide
- Spark Packages
- O'Reilly Radar – Spark articles

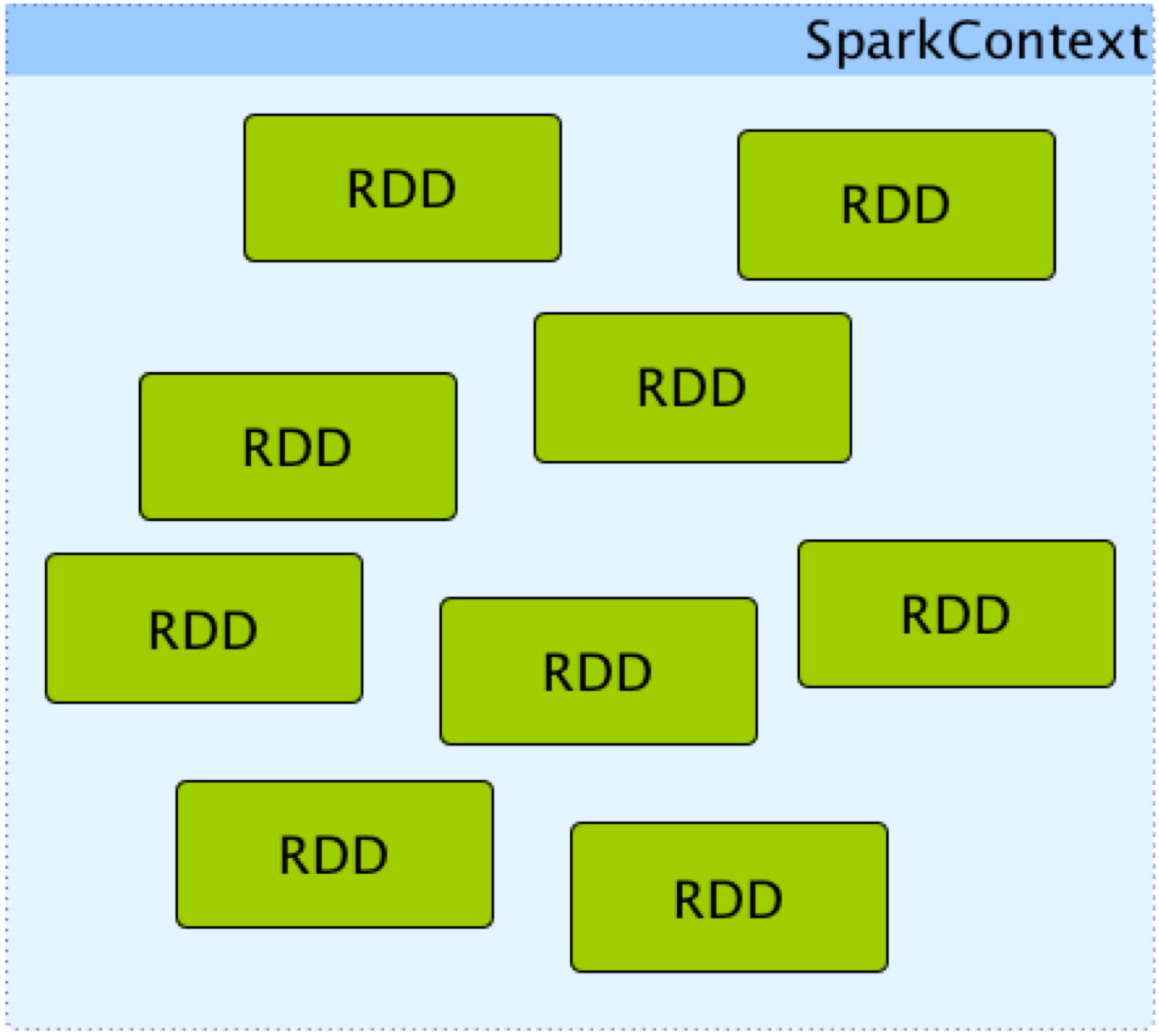
Spark ODBC Driver Download

Spark's ODBC driver lets you connect Business Intelligence (BI) and other third party applications to the Spark SQL server.

GET IT HERE

https://databricks.com/spark/developer-resources

SparkContext and RDDs



Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-sparkcontext.html

Resilient Distributed Dataset (RDD)

Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma,
Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica
University of California, Berkeley

Abstract

We present Resilient Distributed Datasets (RDDs), a distributed memory abstraction that lets programmers perform in-memory computations on large clusters in a fault-tolerant manner. RDDs are motivated by two types of applications that current computing frameworks handle inefficiently: iterative algorithms and interactive data mining tools. In both cases, keeping data in memory can improve performance by an order of magnitude. To achieve fault tolerance efficiently, RDDs provide a restricted form of shared memory, based on coarse-grained transformations rather than fine-grained updates to shared state. However, we show that RDDs are expressive enough to capture a wide class of computations, including recent specialized programming models for iterative jobs, such as Pregel, and new applications that these models do not capture. We have implemented RDDs in a system called Spark, which we evaluate through a variety of user applications and benchmarks.

1 Introduction

Cluster computing frameworks like MapReduce [10] and Dryad [19] have been widely adopted for large-scale data analytics. These systems let users write parallel computation, which can dominate application execution times.

Recognizing this problem, researchers have developed specialized frameworks for some applications that require data reuse. For example, Pregel [22] is a system for iterative graph computations that keeps intermediate data in memory, while HaLoop [7] offers an iterative MapReduce interface. However, these frameworks only support specific computation patterns (*e.g.*, looping a series of MapReduce steps), and perform data sharing implicitly for these patterns. They do not provide abstractions for more general reuse, *e.g.*, to let a user load several datasets into memory and run ad-hoc queries across them.

In this paper, we propose a new abstraction called *resilient distributed datasets (RDDs)* that enables efficient data reuse in a broad range of applications. RDDs are fault-tolerant, parallel data structures that let users explicitly persist intermediate results in memory, control their partitioning to optimize data placement, and manipulate them using a rich set of operators.

The main challenge in designing RDDs is defining a programming interface that can provide fault tolerance *efficiently*. Existing abstractions for in-memory storage on clusters, such as distributed shared memory [24], key-value stores [25], databases, and Piccolo [27], offer an

Source: https://www.cs.berkeley.edu/~matei/papers/2012/nsdi_spark.pdf

RDD

- **R**esilient
  - –fault-tolerant and so able to recompute missing or damaged partitions on node failures with the help of RDD lineage graph.
- **D**istributed across clusters.
- **D**ataset
  - –a collection of partitioned data.

Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-rdd.html

Resilient Distributed Datasets (RDD)

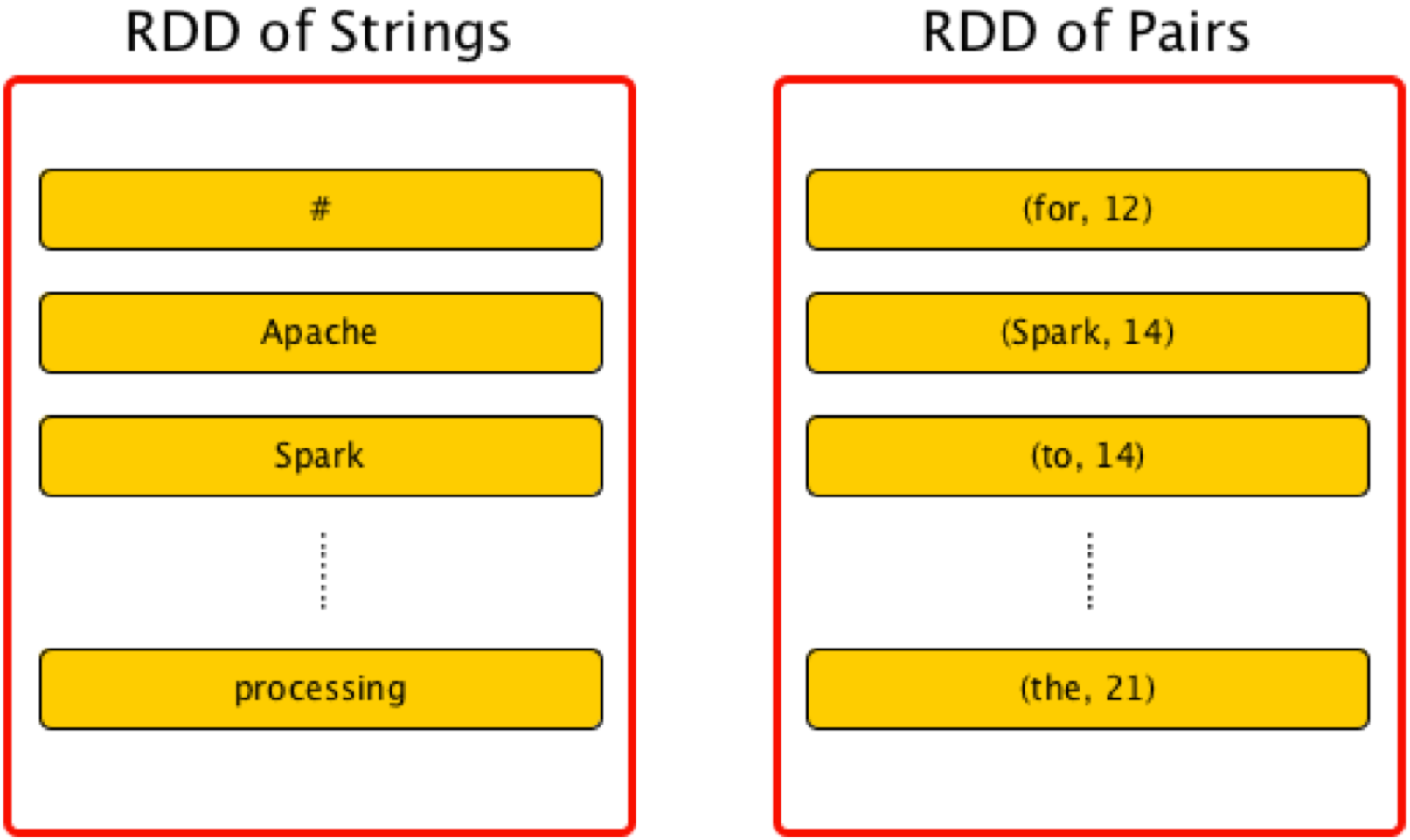
are the primary abstraction in Spark – a fault-tolerant collection of elements that can be operated on in parallel

Source: Databricks (2016), Intro to Apache Spark

Resilient Distributed Dataset (RDD)

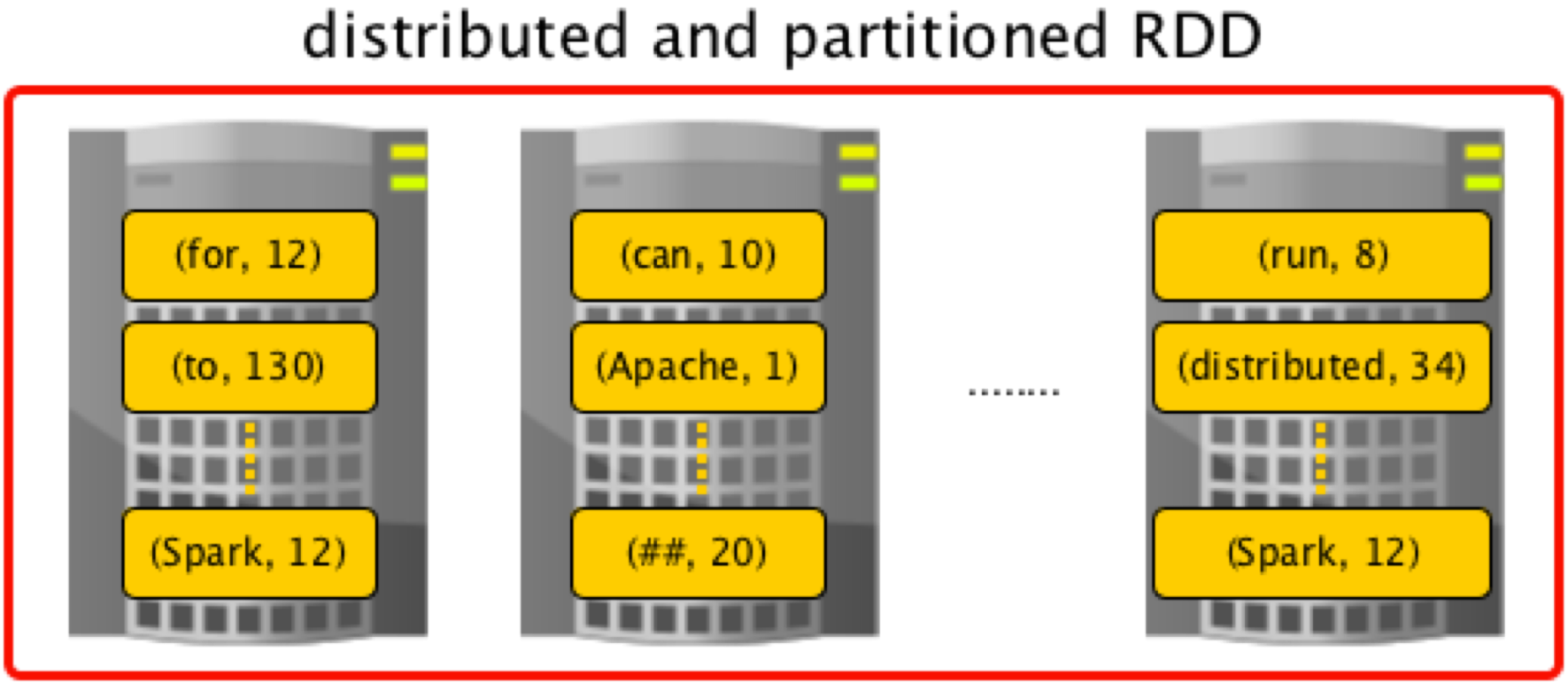
Spark's "Interface" to data

Spark RDD



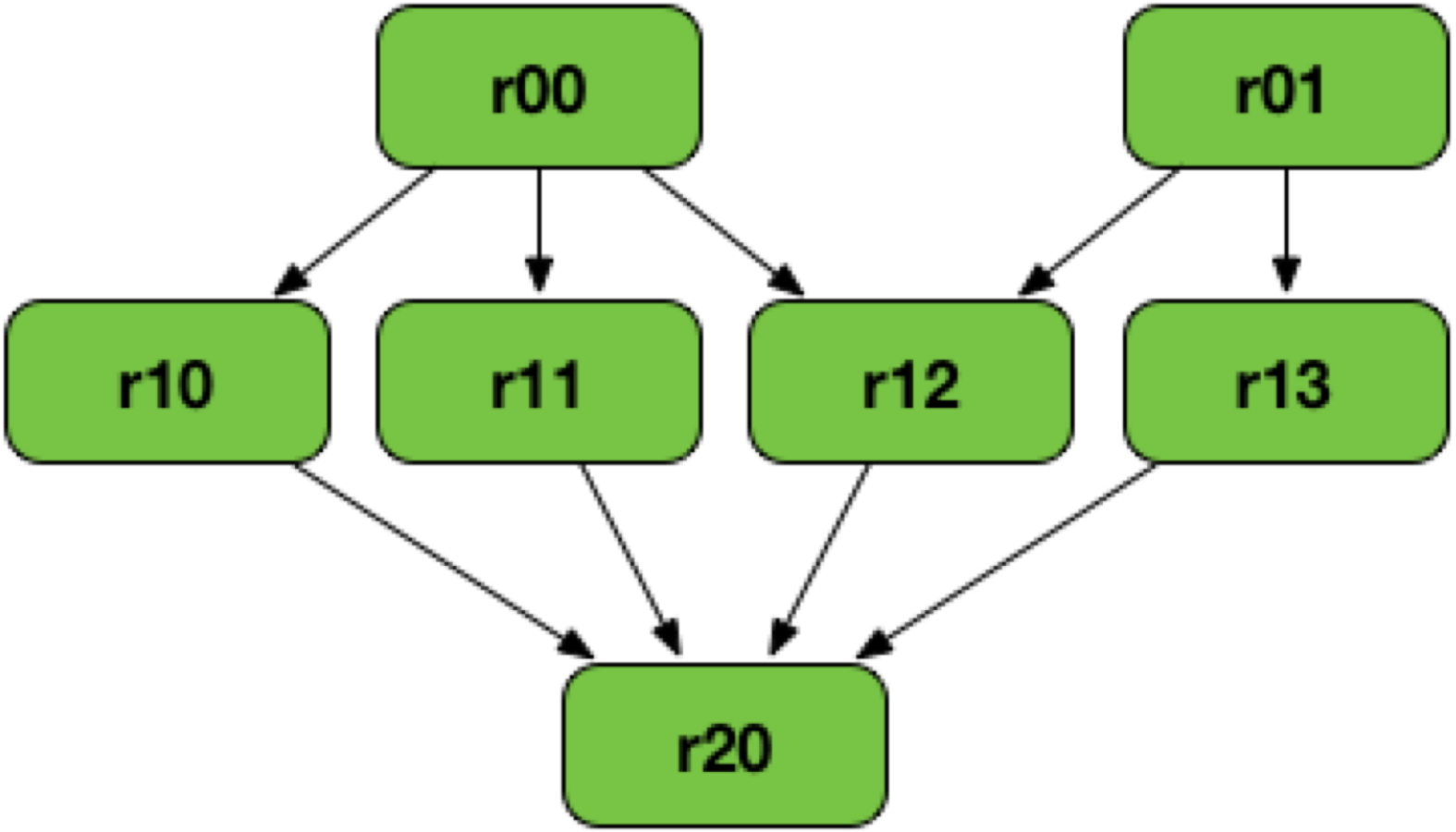
Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-rdd.html

Spark RDD



Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-rdd.html

RDD Lineage Graph
(RDD operator graph)



Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-rdd.html

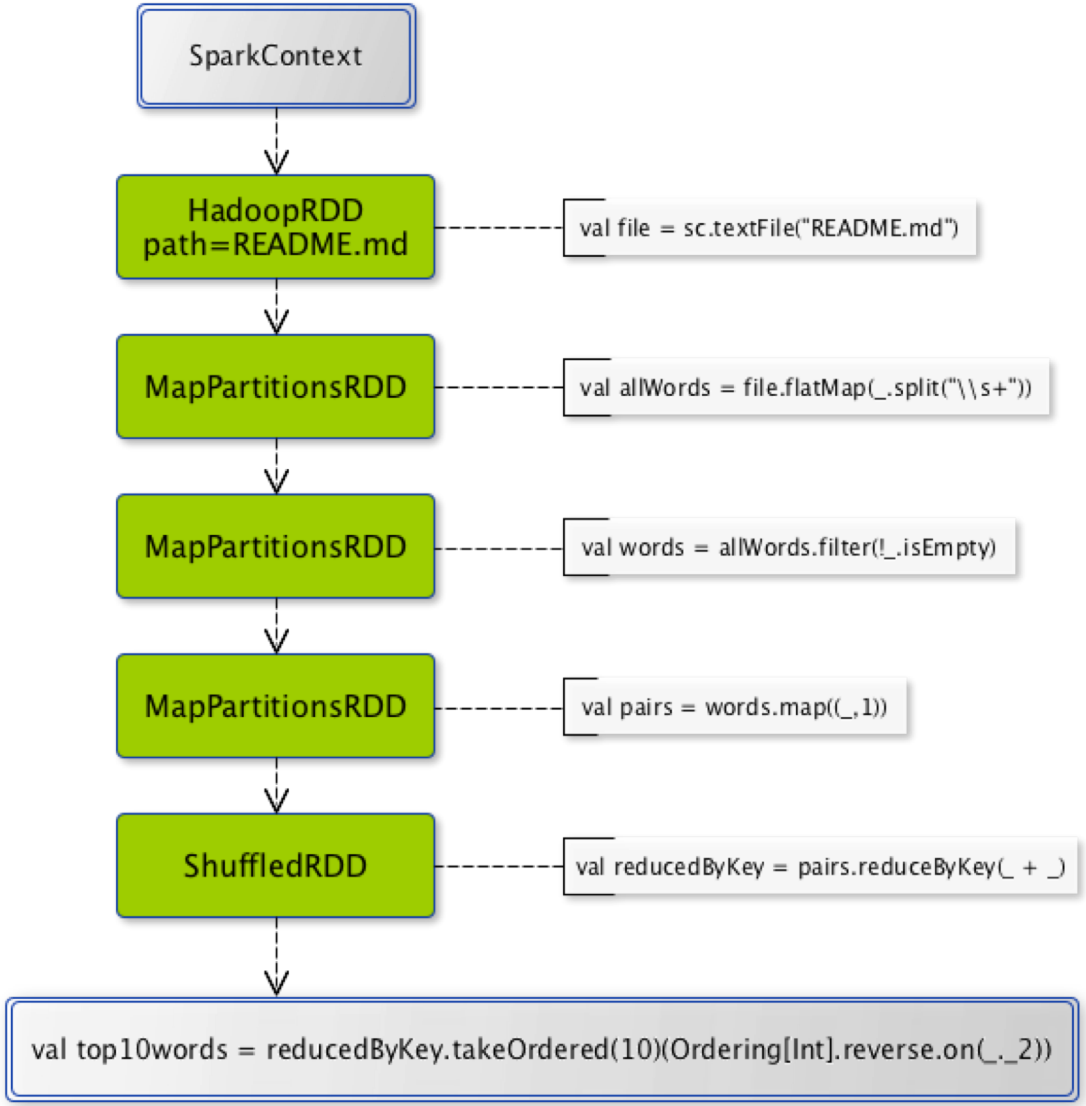
Resilient Distributed Dataset (RDD) Operations

Transformations

Actions

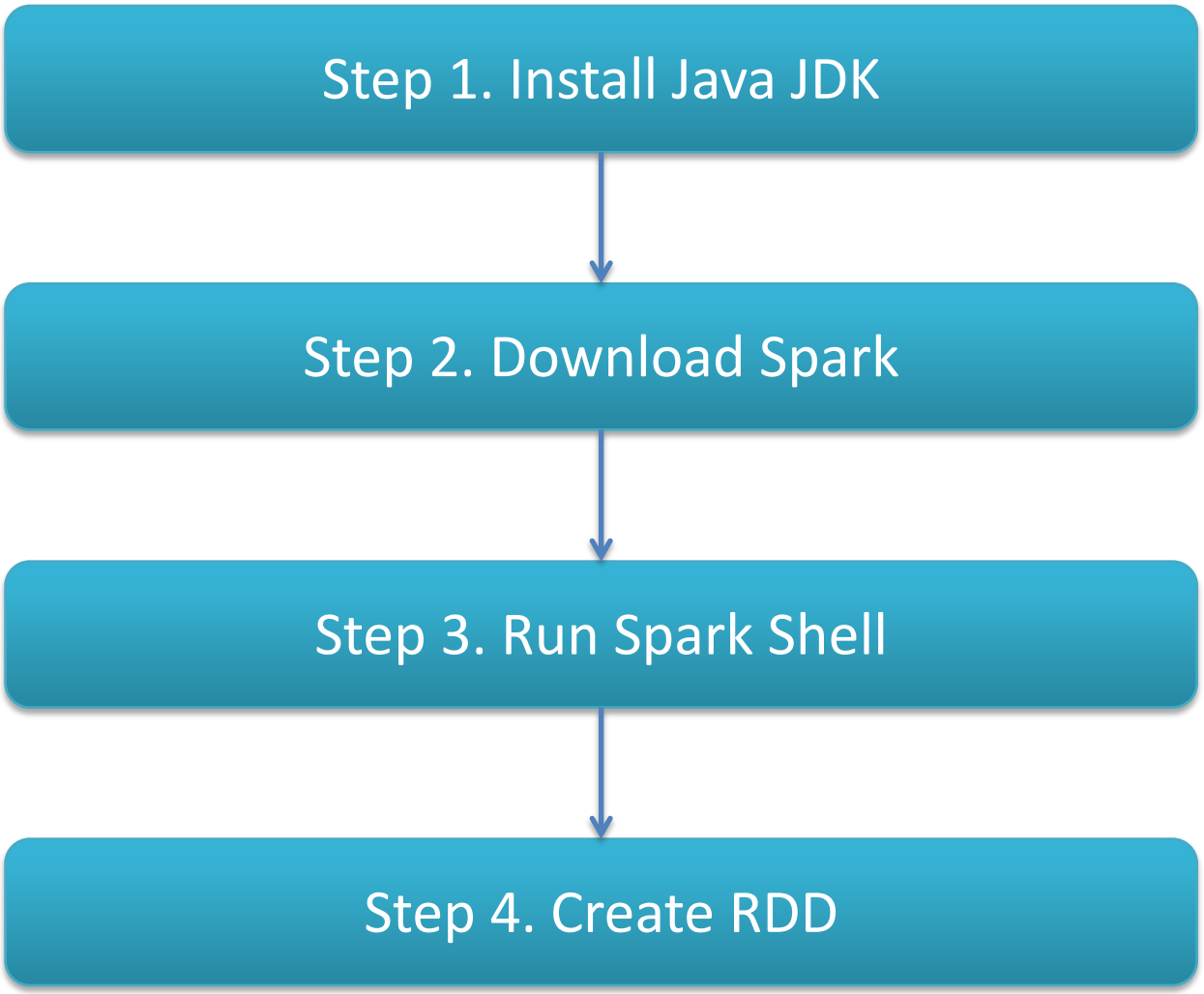
Source: Liondatasystems (2015), Intro to Apache Spark (Brain-Friendly Tutorial)

Spark RDD Operations



Source: https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/spark-rdd-operations.html

Get Started using Apache Spark on a Personal Computer (Windows/Mac OS X)



Source: Databricks (2016), Intro to Apache Spark

Apache Spark

Lightning-fast unified analytics engine

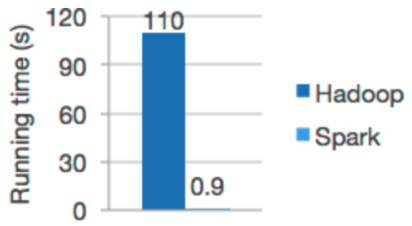
Download | Libraries | Documentation | Examples | Community | Developers | Apache Software Foundation

Apache Spark™ is a unified analytics engine for large-scale data processing.

Speed

Run workloads 100x faster.

Apache Spark achieves high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.



Logistic regression in Hadoop and Spark

Ease of Use

Write applications quickly in Java, Scala, Python, R, and SQL.

Spark offers over 80 high-level operators that make it easy to build parallel

```
df = spark.read.json("logs.json")
df.where("age > 21")
  .select("name.first").show()
```

Spark's Python DataFrame API
Read JSON files with automatic schema inference

Latest News

Spark 2.3.2 released (Sep 24, 2018)

Spark+AI Summit (October 2-4th, 2018, London) agenda posted (Jul 24, 2018)

Spark 2.2.2 released (Jul 02, 2018)

Spark 2.1.3 released (Jun 29, 2018)

Archive

APACHECON North America
September 24-27, 2018
Montréal, Canada

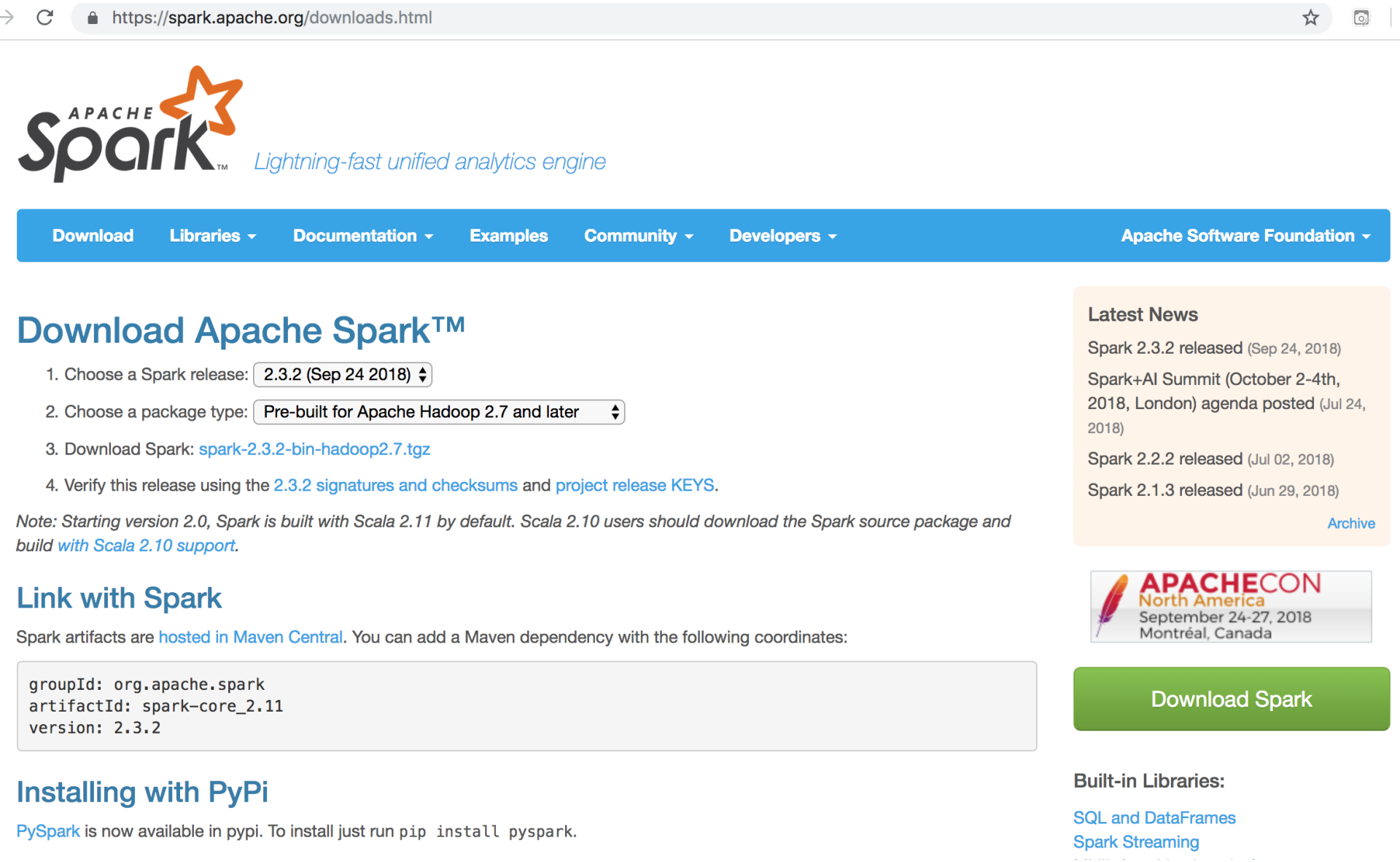
Download Spark

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

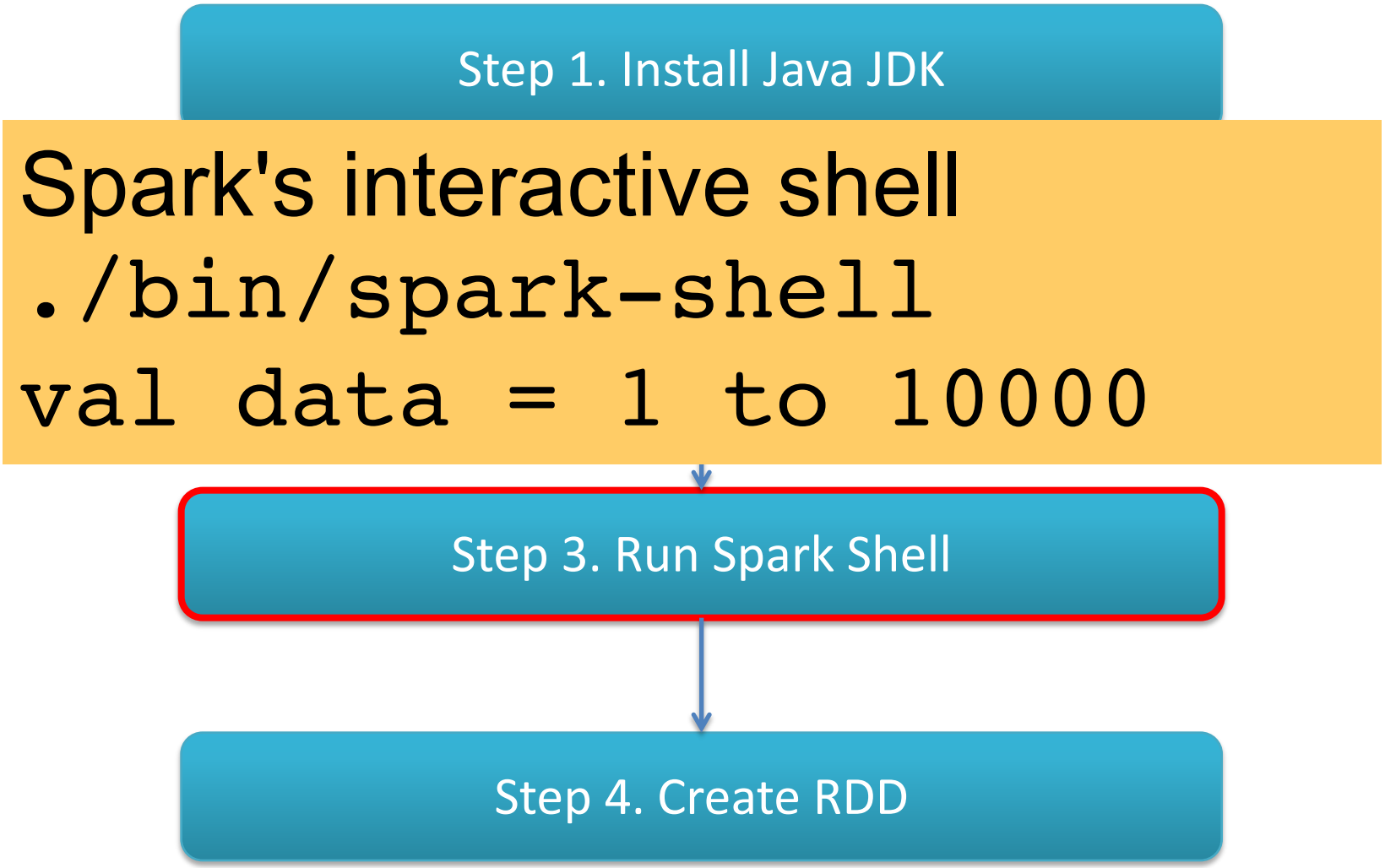
http://spark.apache.org/

Spark Download



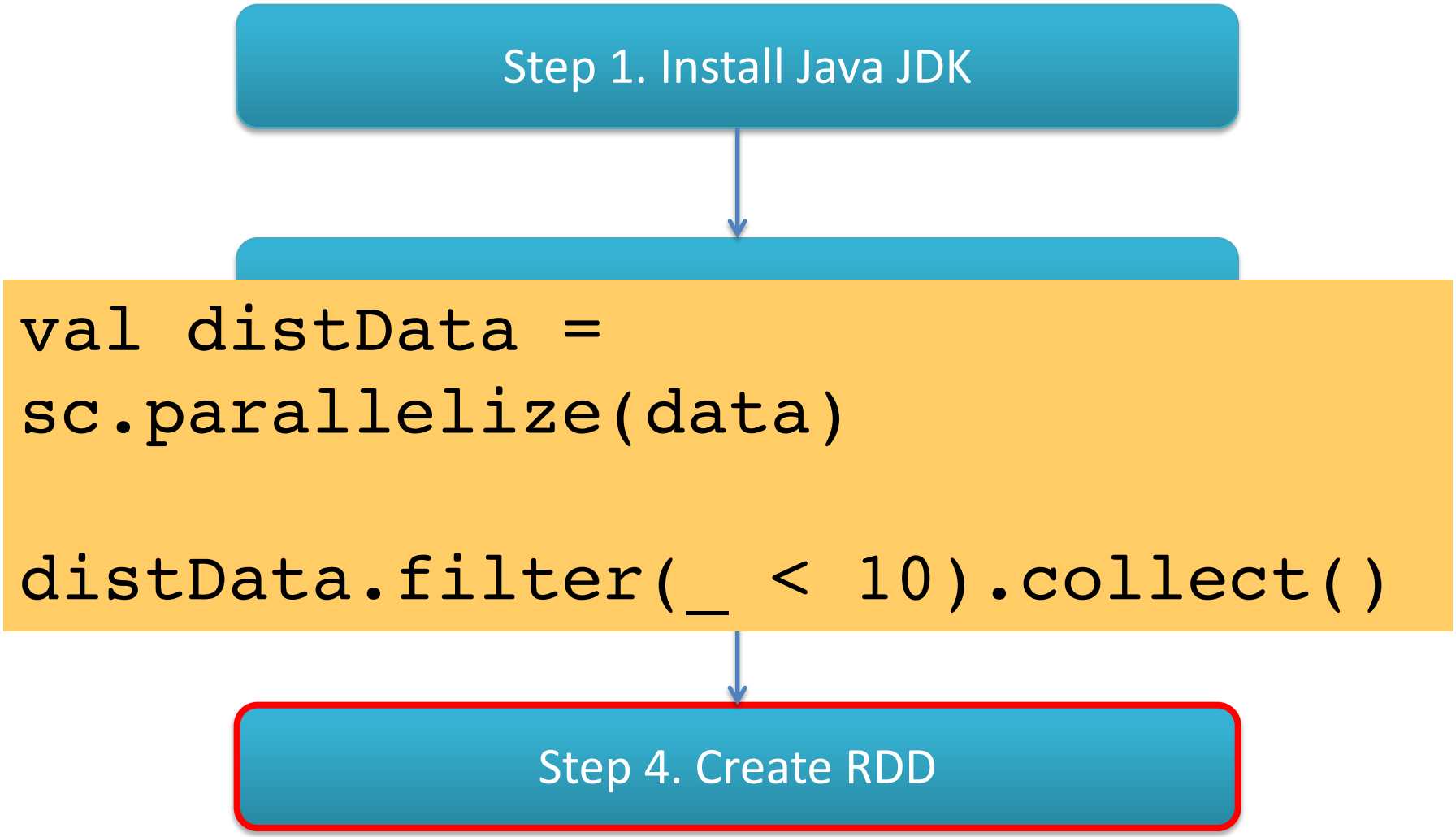
http://spark.apache.org/downloads.html

Get Started using Apache Spark on a Personal Computer (Windows/Mac OS X)



Source: Databricks (2016), Intro to Apache Spark

Get Started using Apache Spark on a Personal Computer (Windows/Mac OS X)



Source: Databricks (2016), Intro to Apache Spark

Word Count

Hello World in Big Data

Spark Examples

Lightning-fast cluster computing

Download | Libraries | Documentation | Examples | Community | FAQ | Apache Software Foundation

Spark Examples

These examples give a quick overview of the Spark API. Spark is built on the concept of *distributed datasets*, which contain arbitrary Java or Python objects. You create a dataset from external data, then apply parallel operations to it. The building block of the Spark API is its RDD API. In the RDD API, there are two types of operations: *transformations*, which define a new dataset based on previous ones, and *actions*, which kick off a job to execute on a cluster. On top of Spark's RDD API, high level APIs are provided, e.g. DataFrame API and Machine Learning API. These high level APIs provide a concise way to conduct certain data operations. In this page, we will show examples using RDD API as well as examples using high level APIs.

RDD API Examples

Word Count

In this example, we use a few transformations to build a dataset of (String, Int) pairs called `counts` and then save it to a file.

Python | Scala | Java

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

Latest News

Submission is open for Spark Summit San Francisco (Feb 11, 2016)

Spark Summit East (Feb 16, 2016, New York) agenda posted (Jan 14, 2016)

Spark 1.6.0 released (Jan 04, 2016)

CFP for Spark Summit East 2016 is closing soon! (Nov 19, 2015)

Archive

Download Spark

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Packages

Source: http://spark.apache.org/examples.html

Spark Word Count Python

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

Source: http://spark.apache.org/examples.html

Spark Word Count
Scala

```
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
                 .map(word => (word, 1))
                 .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

Source: http://spark.apache.org/examples.html

Spark Word Count
Java

```
JavaRDD<String> textFile = sc.textFile("hdfs://...");
JavaRDD<String> words = textFile.flatMap(new FlatMapFunction<String, String>()
{
  public Iterable<String> call(String s) { return Arrays.asList(s.split(" "));
}
});
JavaPairRDD<String, Integer> pairs = words.mapToPair(new PairFunction<String,
String, Integer>() {
  public Tuple2<String, Integer> call(String s) { return new Tuple2<String,
Integer>(s, 1); }
});
JavaPairRDD<String, Integer> counts = pairs.reduceByKey(new Function2<Integer,
Integer, Integer>() {
  public Integer call(Integer a, Integer b) { return a + b; }
});
counts.saveAsTextFile("hdfs://...");
```

Source: http://spark.apache.org/examples.html

Spark Word Count

Python

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

Scala

```
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
                 .map(word => (word, 1))
                 .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

Java

```
JavaRDD<String> textFile = sc.textFile("hdfs://...");
JavaRDD<String> words = textFile.flatMap(new FlatMapFunction<String, String>() {
  public Iterable<String> call(String s) { return Arrays.asList(s.split(" ")); }
});
JavaPairRDD<String, Integer> pairs = words.mapToPair(new PairFunction<String, String, Integer>() {
  public Tuple2<String, Integer> call(String s) { return new Tuple2<String, Integer>(s, 1); }
});
JavaPairRDD<String, Integer> counts = pairs.reduceByKey(new Function2<Integer, Integer, Integer>() {
  public Integer call(Integer a, Integer b) { return a + b; }
});
counts.saveAsTextFile("hdfs://...");
```

Source: http://spark.apache.org/examples.html

Spark Word Count Python

```
text_file = sc.textFile(”input_readme.txt")
counts = text_file.flatMap(lambda line: line.split(" "))
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile(”output_wordcount.txt")
```

Source: http://spark.apache.org/examples.html

WordCount Example

- Input

```
Hello World Bye World
Hello Hadoop Goodbye Hadoop
```

- For the given sample input the map emits

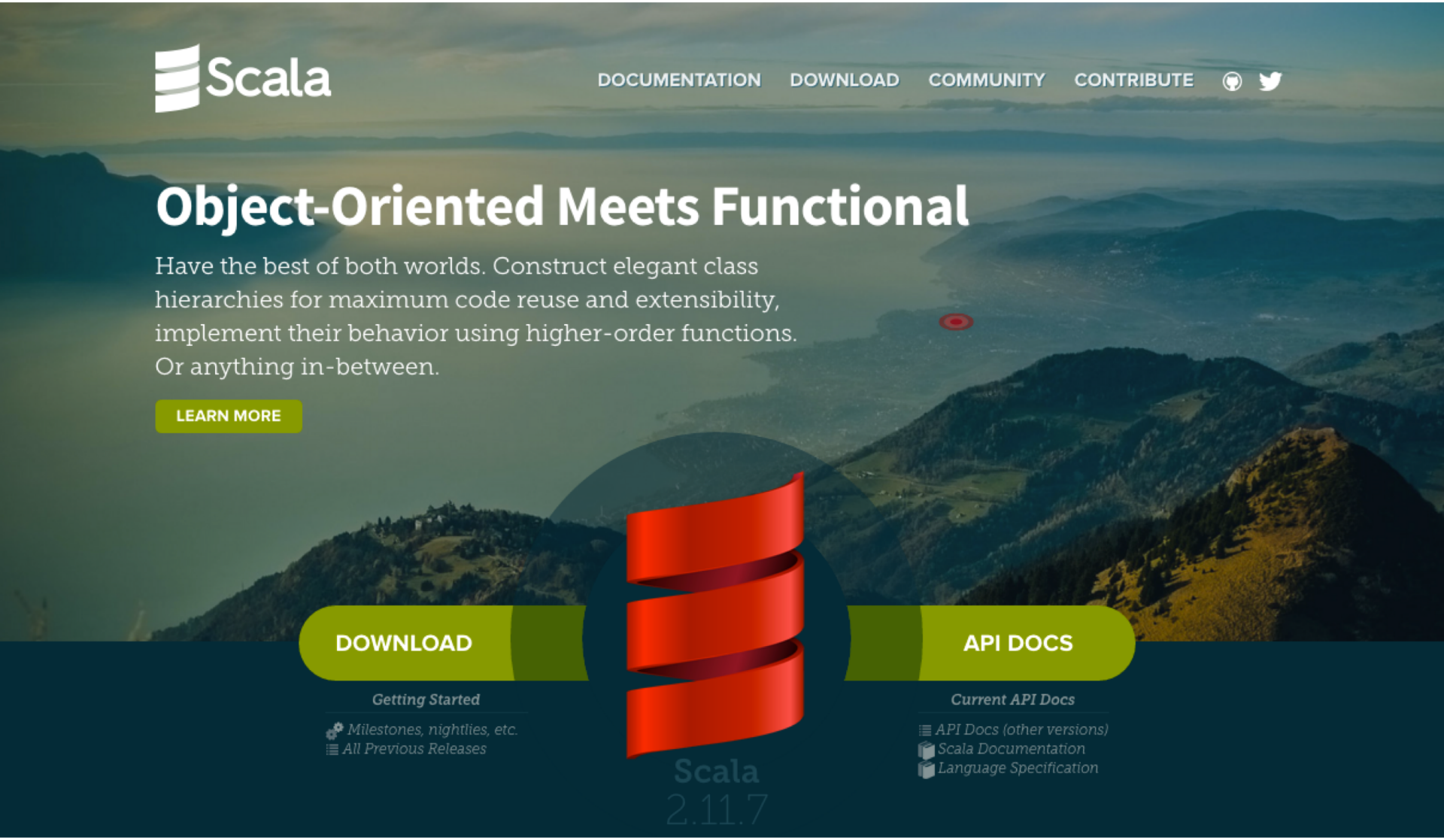
```
< Hello, 1>
< World, 1>
< Bye, 1>
< World, 1>
< Hello, 1>
< Hadoop, 1>
< Goodbye, 1>
< Hadoop, 1>
```

-

```
< Bye, 1>
< Goodbye, 1>
< Hadoop, 2>
< Hello, 2>
< World, 2>
```

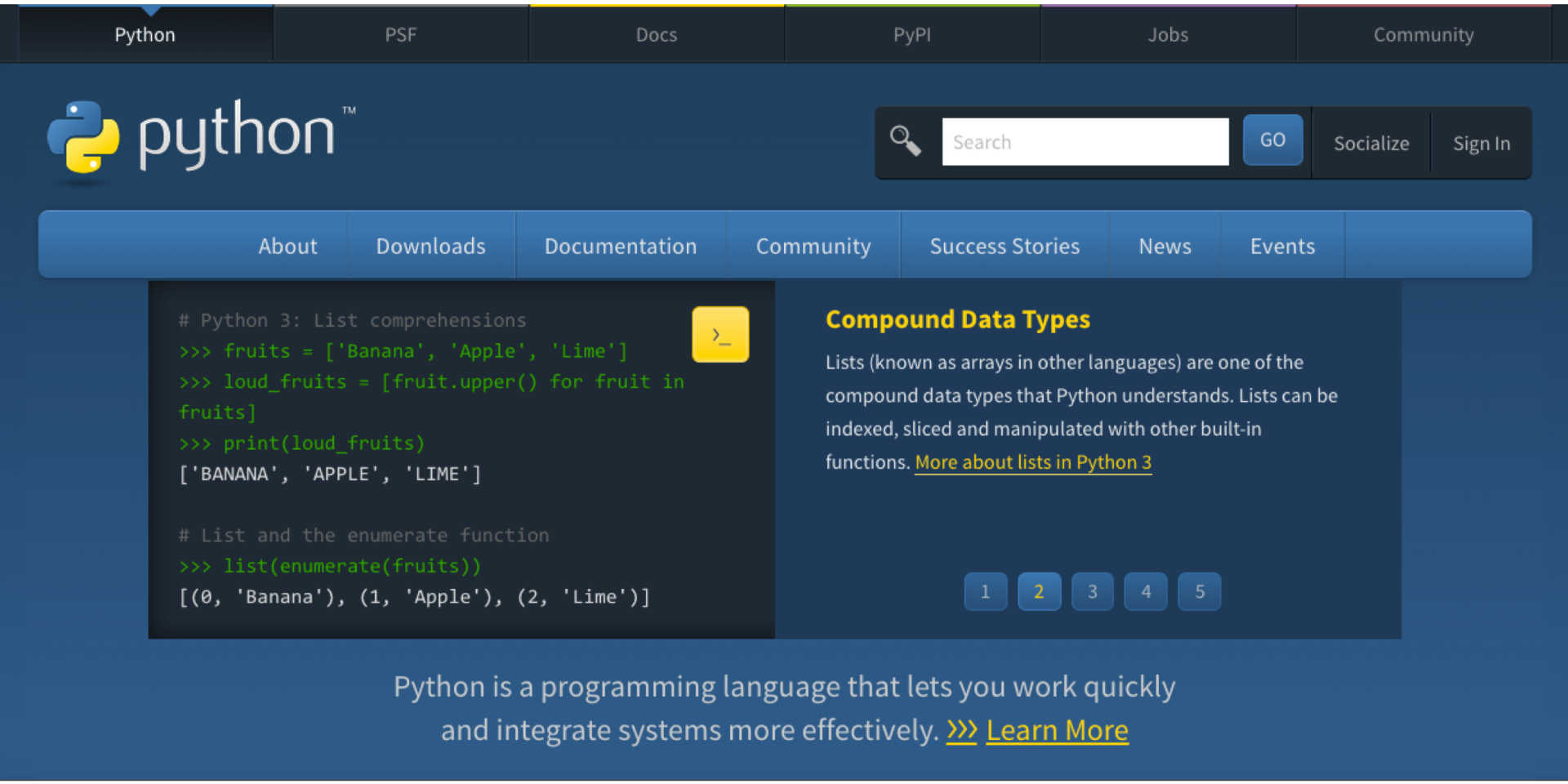
Source: http://www.slideshare.net/tcloudcomputing-tw/hadoop-product-family-2013q3-t-cloud-computing-26218439

Scala



http://www.scala-lang.org/

Python



https://www.python.org/

PySpark: Spark Python API

Next topic

pyspark package

This Page

Show Source

Quick search

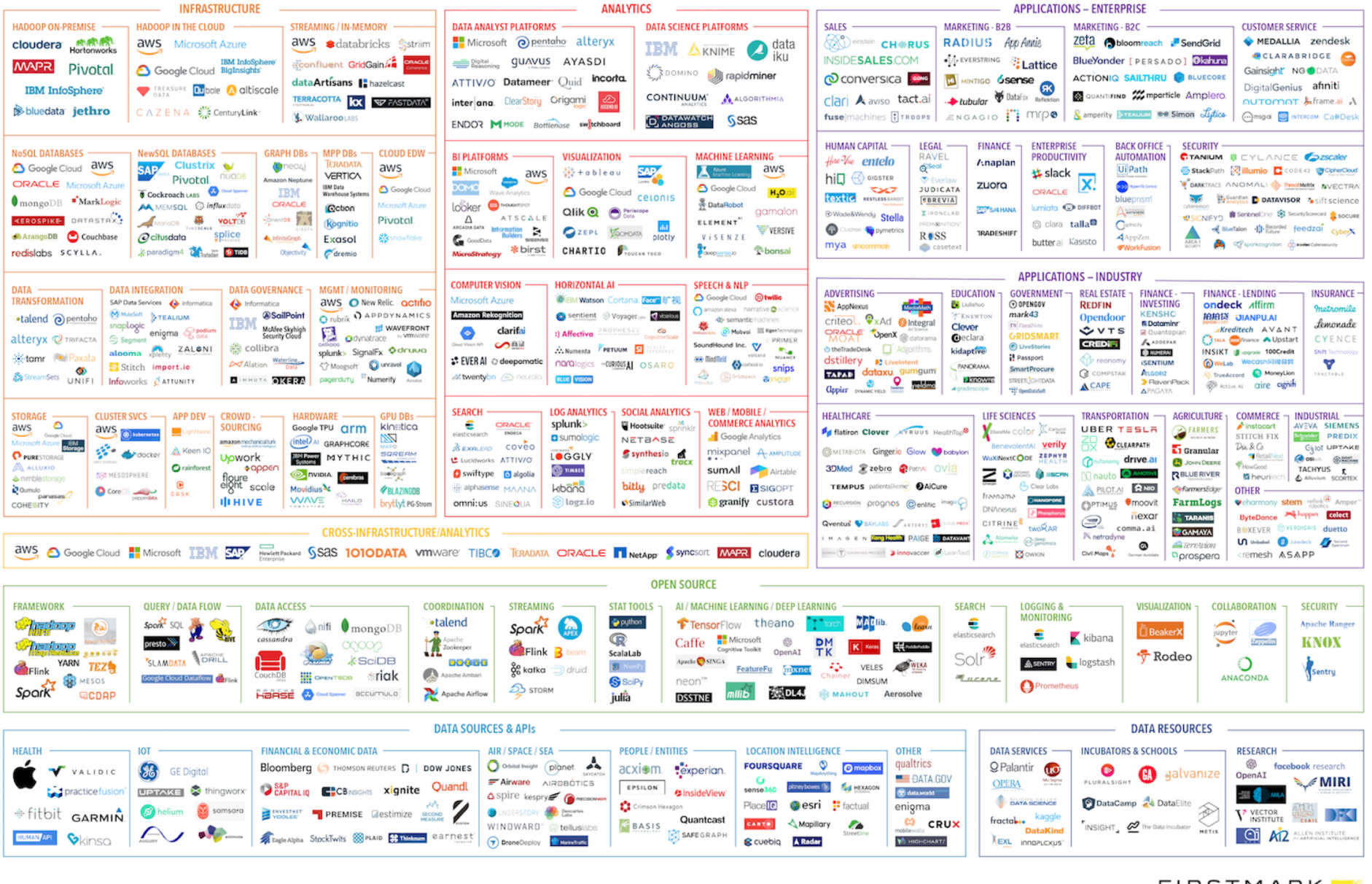
Go

Welcome to Spark Python API Docs!

Contents:

http://spark.apache.org/docs/latest/api/python/

Big Data & AI Landscape 2018



Final 2018 version, updated 07/15/2018

© Matt Turck (@mattturck), Demi Obayomi (@demi_obayomi), & FirstMark (@firstmarkcap)

mattturck.com/bigdata2018

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

Source: http://mattturck.com/bigdata2018/

References

- EMC Education Services (2015),
  Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley
- Shiva Achari (2015),
  Hadoop Essentials - Tackling the Challenges of Big Data with Hadoop, Packt Publishing
- Mike Frampton (2015),
  Mastering Apache Spark, Packt Publishing
- Deepak Ramanathan (2014),
  SAS Modernization architectures - Big Data Analytics,
  http://www.slideshare.net/deepakramanathan/sas-modernization-architectures-big-data-analytics
- Spark Apache (2018), PySpark: Spark Python API,
  http://spark.apache.org/docs/latest/api/python/