

Social Computing and Big Data Analytics

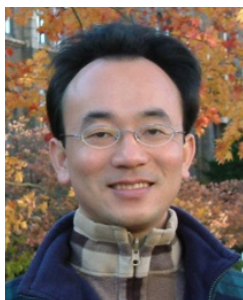
社群運算與大數據分析

Big Data Processing Platforms with SMACK: Spark, Mesos, Akka, Cassandra and Kafka (大數據處理平台SMACK)

1052SCBDA04

MIS MBA (M2226) (8606)

Wed, 8,9, (15:10-17:00) (L206)



Min-Yuh Day

戴敏育

Assistant Professor

專任助理教授

Dept. of Information Management, Tamkang University

淡江大學 資訊管理學系

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

2017-03-08



課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2017/02/15	Course Orientation for Social Computing and Big Data Analytics (社群運算與大數據分析課程介紹)
2	2017/02/22	Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data (資料科學與大數據分析： 探索、分析、視覺化與呈現資料)
3	2017/03/01	Fundamental Big Data: MapReduce Paradigm, Hadoop and Spark Ecosystem (大數據基礎：MapReduce典範、 Hadoop與Spark生態系統)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
4	2017/03/08	Big Data Processing Platforms with SMACK: Spark, Mesos, Akka, Cassandra and Kafka (大數據處理平台SMACK： Spark, Mesos, Akka, Cassandra, Kafka)
5	2017/03/15	Big Data Analytics with Numpy in Python (Python Numpy 大數據分析)
6	2017/03/22	Finance Big Data Analytics with Pandas in Python (Python Pandas 財務大數據分析)
7	2017/03/29	Text Mining Techniques and Natural Language Processing (文字探勘分析技術與自然語言處理)
8	2017/04/05	Off-campus study (教學行政觀摩日)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
9	2017/04/12	Social Media Marketing Analytics (社群媒體行銷分析)
10	2017/04/19	期中報告 (Midterm Project Report)
11	2017/04/26	Deep Learning with Theano and Keras in Python (Python Theano 和 Keras 深度學習)
12	2017/05/03	Deep Learning with Google TensorFlow (Google TensorFlow 深度學習)
13	2017/05/10	Sentiment Analysis on Social Media with Deep Learning (深度學習社群媒體情感分析)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
14	2017/05/17	Social Network Analysis (社會網絡分析)
15	2017/05/24	Measurements of Social Network (社會網絡量測)
16	2017/05/31	Tools of Social Network Analysis (社會網絡分析工具)
17	2017/06/07	Final Project Presentation I (期末報告 I)
18	2017/06/14	Final Project Presentation II (期末報告 II)

2017/03/08

Big Data Processing Platforms with SMACK:

**Spark, Mesos, Akka,
Cassandra and Kafka**

(大數據處理平台 SMACK :

**Spark, Mesos, Akka,
Cassandra, Kafka)**

SMACK Architectures

Building data processing platforms with
Spark, **M**esos, **A**kka, **C**assandra and **K**afka

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

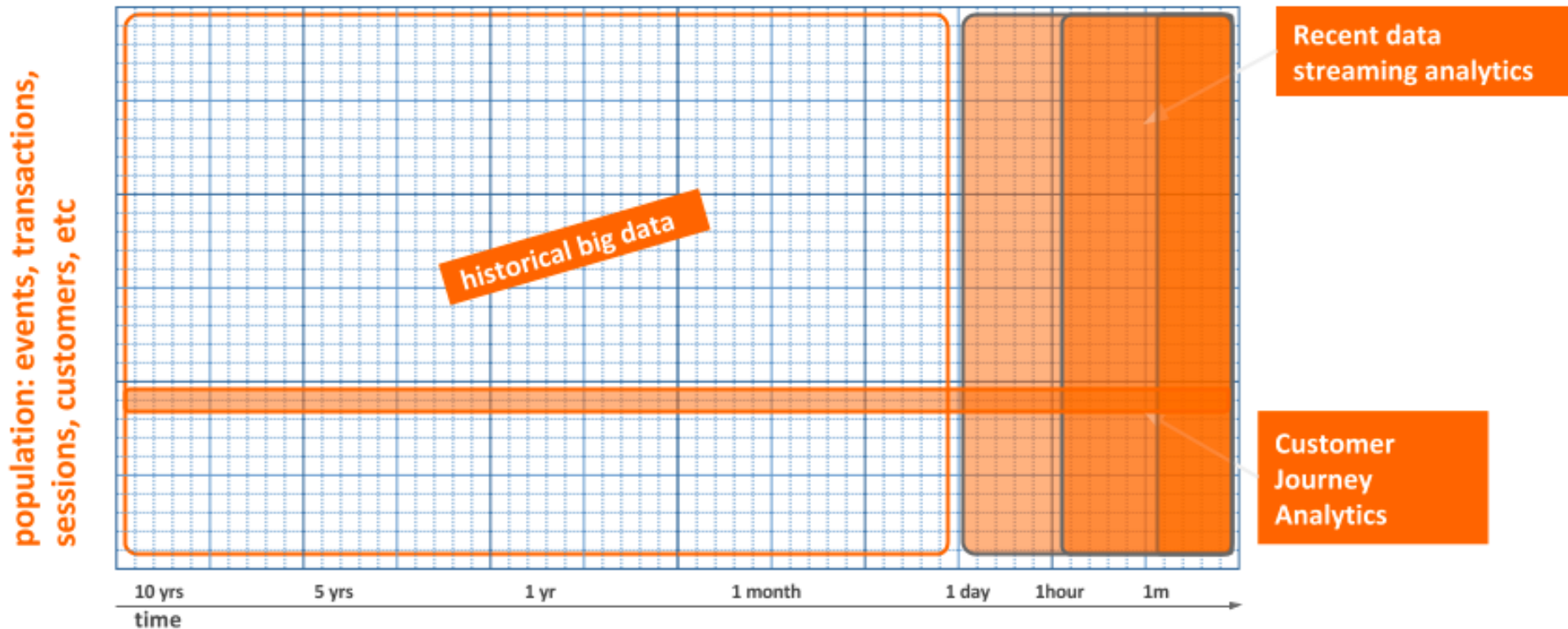
- **Kafka**



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

Why SMACK?

Big, Fast and Event Data Processing



S  Spark

M  MESOS

A  akka

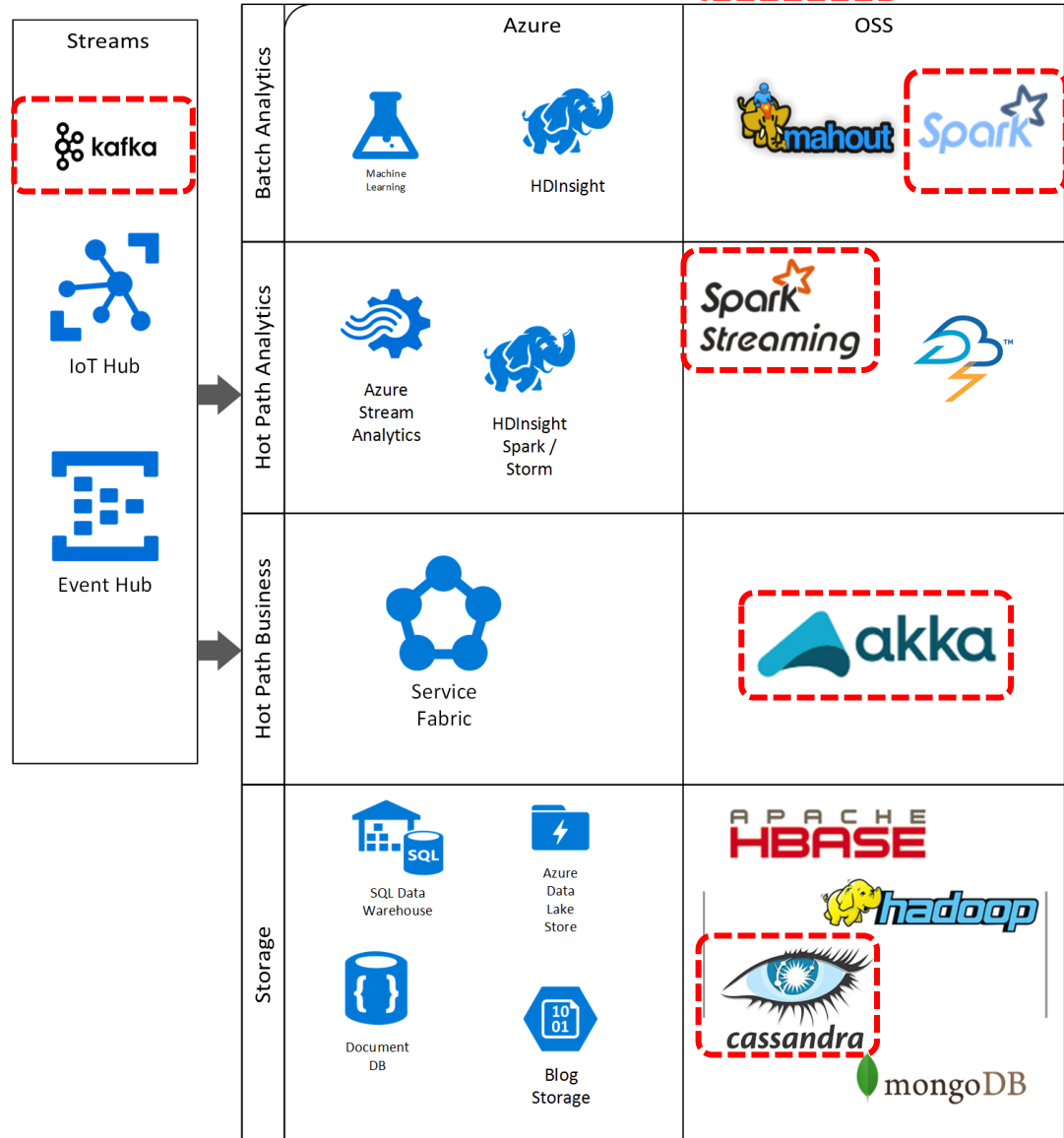
C  cassandra

K  kafka

Managed by



Azure Container Services



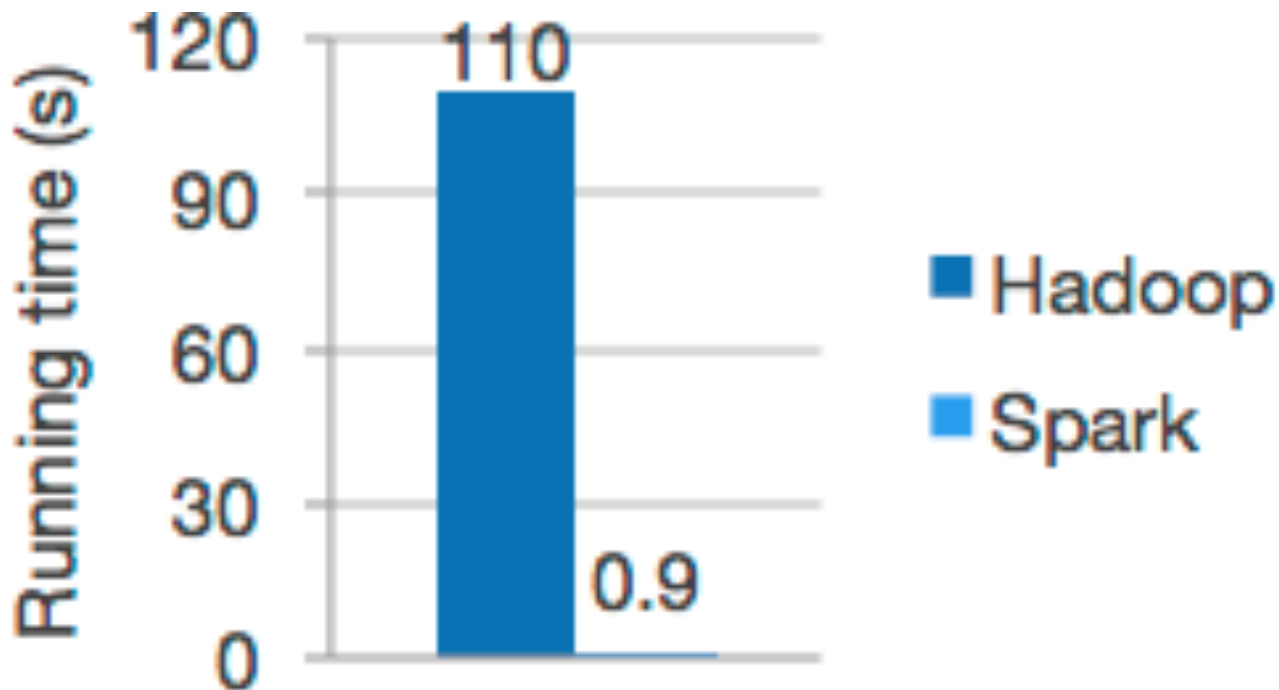
Spark Ecosystem



Lightning-fast cluster computing

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

Logistic regression in Hadoop and Spark



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

Ease of Use

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



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)
```

Spark and Hadoop





Spark Ecosystem

Spark
SQL

Spark
Streaming

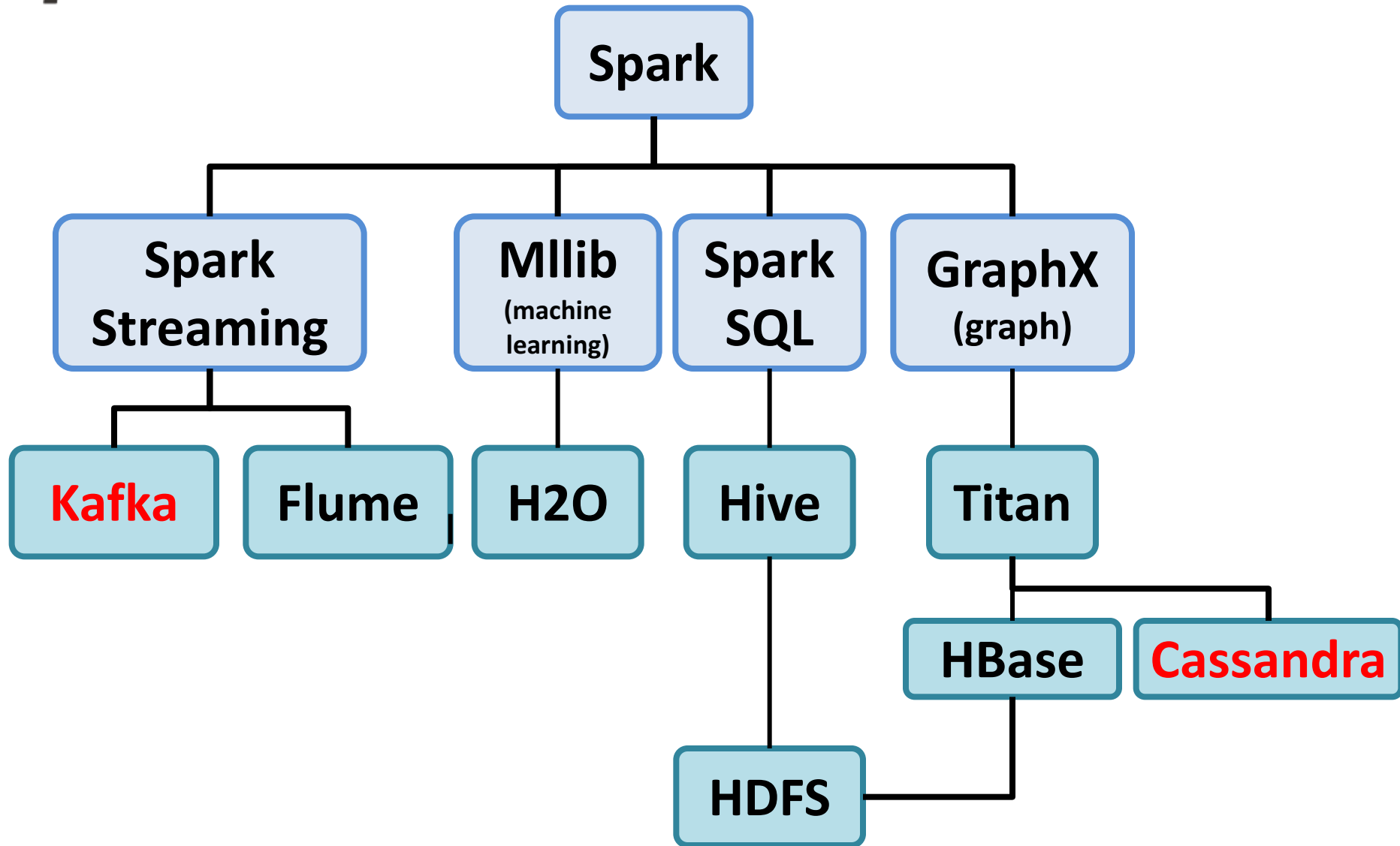
MLlib
(machine
learning)

GraphX
(graph)

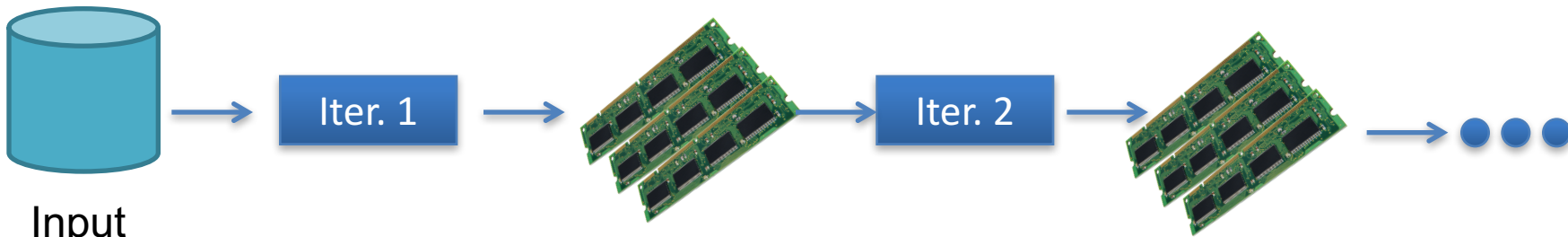
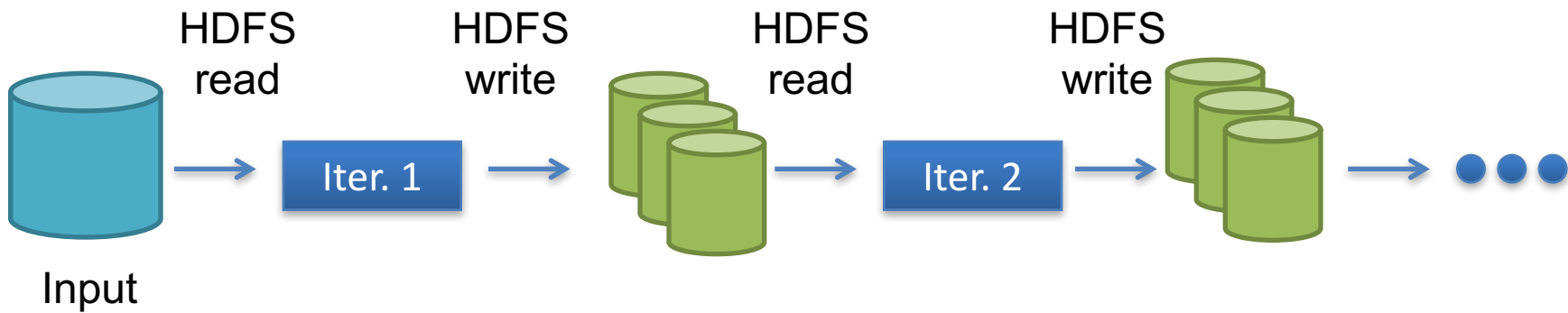
Apache Spark



Spark Ecosystem



Hadoop vs. Spark



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

Spark

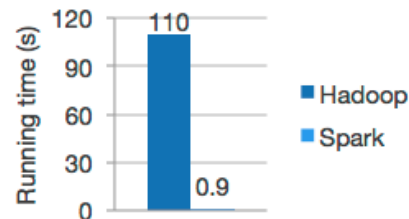
[Download](#)[Libraries ▾](#)[Documentation ▾](#)[Examples](#)[Community ▾](#)[FAQ](#)[Apache Software Foundation ▾](#)

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

Speed

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

Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.



Logistic regression in Hadoop and Spark

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](#)

Ease of Use

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

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala, Python and R shells.

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

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

Word count in Spark's Python API

Built-in Libraries:

[SQL and DataFrames](#)[Spark Streaming](#)[MLlib \(machine learning\)](#)[GraphX \(graph\)](#)[Third-Party Packages](#)

Mesos

Apache Software Foundation ▾ / Apache Mesos



Getting Started

Documentation

Downloads

Community

Program against your datacenter like it's a single pool of resources

Apache Mesos abstracts CPU, memory, storage, and other compute resources away from machines (physical or virtual), enabling fault-tolerant and elastic distributed systems to easily be built and run effectively.

⬇ Download [Mesos 0.27.2](#) or learn how to [get started](#)

What is Mesos?

A distributed systems kernel

Mesos is built using the same principles as the Linux kernel, only at a different level of abstraction. The Mesos kernel runs on every machine and provides applications (e.g., Hadoop, Spark, Kafka, Elastic Search) with API's for resource management and scheduling across entire datacenter and cloud environments.

Project Features

- Scalability to 10,000s of nodes

News

- *March 7, 2016* - Mesos 0.27.2 is released! See the [CHANGELOG](#) and [blog post](#) for details.
- *February 22, 2016* - Mesos 0.27.1 is released! See the [CHANGELOG](#) and [blog post](#) for details.
- *February 12, 2016* - MesosCon 2016 CFP is now open! See the [blog post](#) for details.
- *January 31, 2016* - Mesos 0.27.0 is released! See the [CHANGELOG](#) and [blog post](#) for details.
- *December 16, 2015* - Mesos 0.26.0 is released!

<http://mesos.apache.org/>

Akka

[DOCUMENTATION](#)[DOWNLOAD](#)[GET INVOLVED](#)

We Are Reactive



Build powerful concurrent & distributed applications more easily.

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

The power of Akka is also available on the .NET Framework and Mono via the Akka.NET project.

Simple Concurrency & Distribution

Asynchronous and Distributed by Design. High-level abstractions like Actors, Streams and Futures.

Resilient by Design

Write systems that self-heal. Remote and local supervisor hierarchies.



High Performance

50 million msg/sec on a single machine. Small memory footprint; ~2.5 million actors per GB of heap.

Elastic & Decentralized

Adaptive cluster management, load balancing, routing, partitioning and sharding.

Extensible

Use Akka Extensions to adapt Akka to fit your needs.

Cassandra



[Home](#) [Download](#) [Getting Started](#) [Planet Cassandra](#) [Contribute](#)

Welcome

Video

Slides

Welcome to Apache Cassandra™

The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. [Linear scalability](#) and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's support for replicating across multiple datacenters is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive regional outages.

Cassandra's data model offers the convenience of [column indexes](#) with the performance of log-structured updates, strong support for [denormalization](#) and [materialized views](#), and powerful built-in caching.

Download

[Tick-Tock](#) release **3.4** ([Changes](#))

3.0.x release **3.0.4** ([Changes](#))

2.2.x release **2.2.5** ([Changes](#))



[Download options](#)

Overview

Proven

Cassandra is in use at [Constant Contact](#), [CERN](#), [Comcast](#), [eBay](#), [GitHub](#), [GoDaddy](#), [Hulu](#), [Instagram](#), [Intuit](#), [Netflix](#), [Reddit](#), [The](#)

Performant

Cassandra [consistently outperforms](#) popular NoSQL alternatives in benchmarks and [real applications](#), primarily because of

You're in Control

Choose between synchronous or asynchronous replication for each update. Highly available asynchronous operations are

<http://cassandra.apache.org/>

Kafka



- [download](#)
- [introduction](#)
- [uses](#)
- [documentation](#)
- [quickstart](#)
- [performance](#)
- [clients](#)
- [ecosystem](#)
- [faq](#)
- [project](#)
 - [twitter](#)
 - [wiki](#)
 - [bugs](#)
 - [mailing lists](#)
 - [committers](#)
 - [powered by](#)
 - [papers & talks](#)
- [developers](#)
 - [code](#)
 - [projects](#)
 - [contributing](#)
 - [coding guide](#)

Apache Kafka is publish-subscribe messaging rethought as a distributed commit log.

Fast

A single Kafka broker can handle hundreds of megabytes of reads and writes per second from thousands of clients.

Scalable

Kafka is designed to allow a single cluster to serve as the central data backbone for a large organization. It can be elastically and transparently expanded without downtime. Data streams are partitioned and spread over a cluster of machines to allow data streams larger than the capability of any single machine and to allow clusters of co-ordinated consumers

Durable

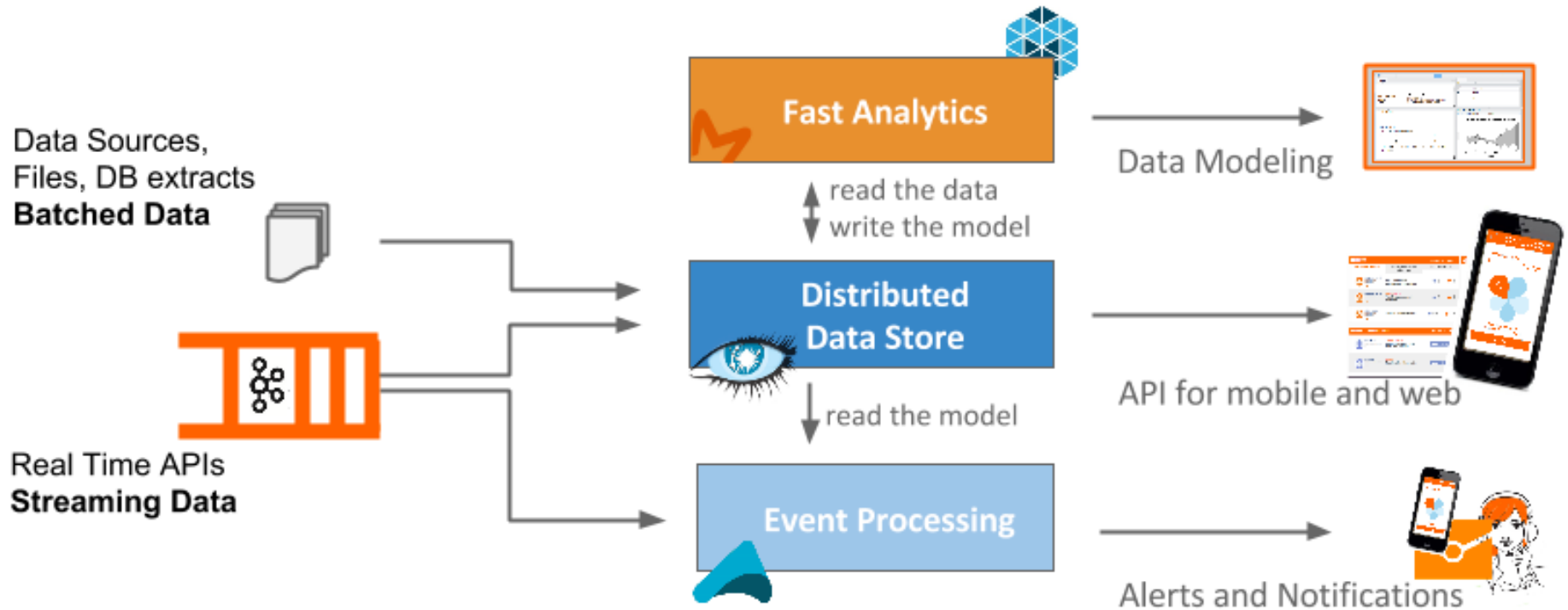
Messages are persisted on disk and replicated within the cluster to prevent data loss. Each broker can handle terabytes of messages without performance impact.

Distributed by Design

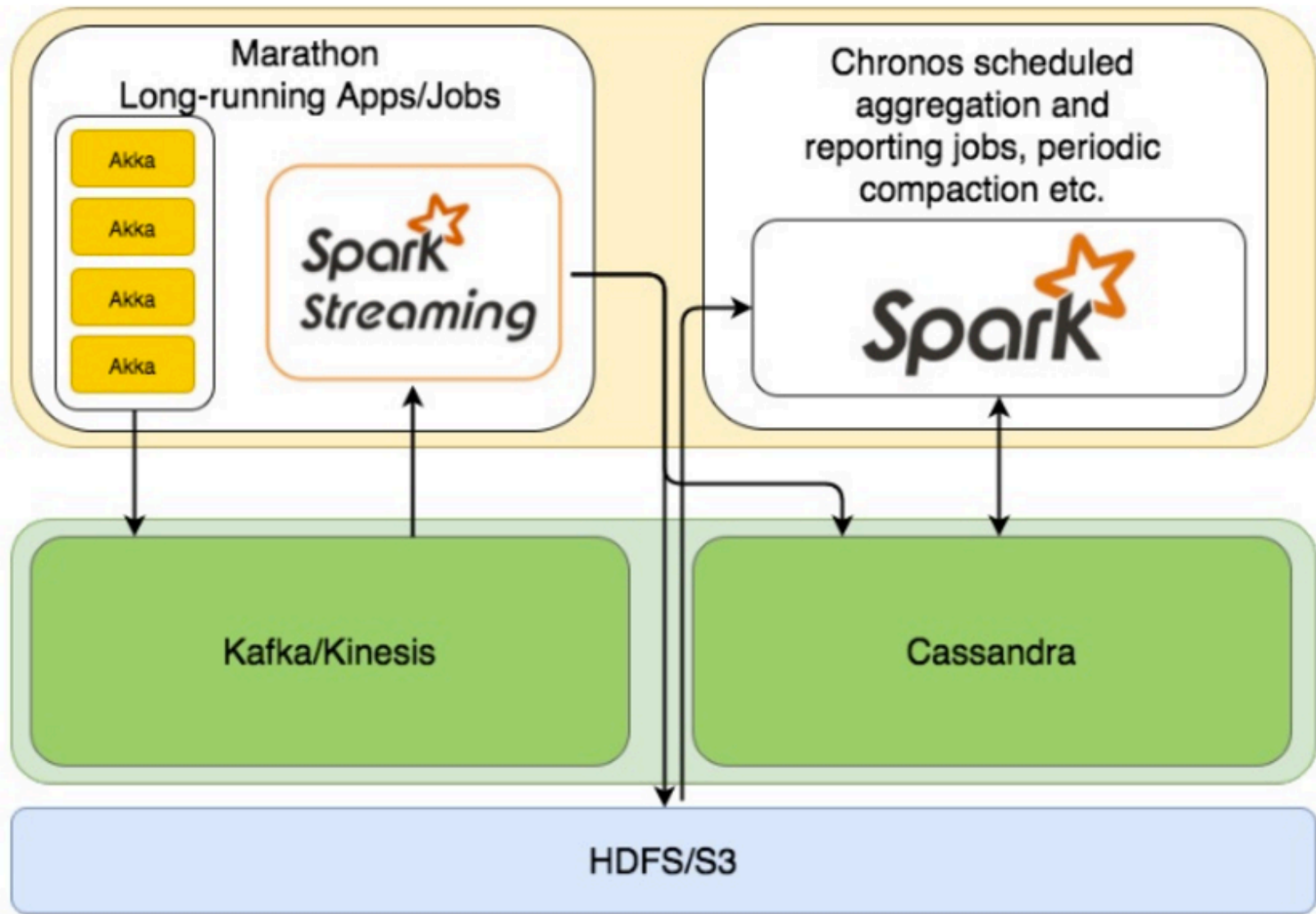
Kafka has a modern cluster-centric design that offers strong durability and fault-tolerance guarantees.

SMACK Stack

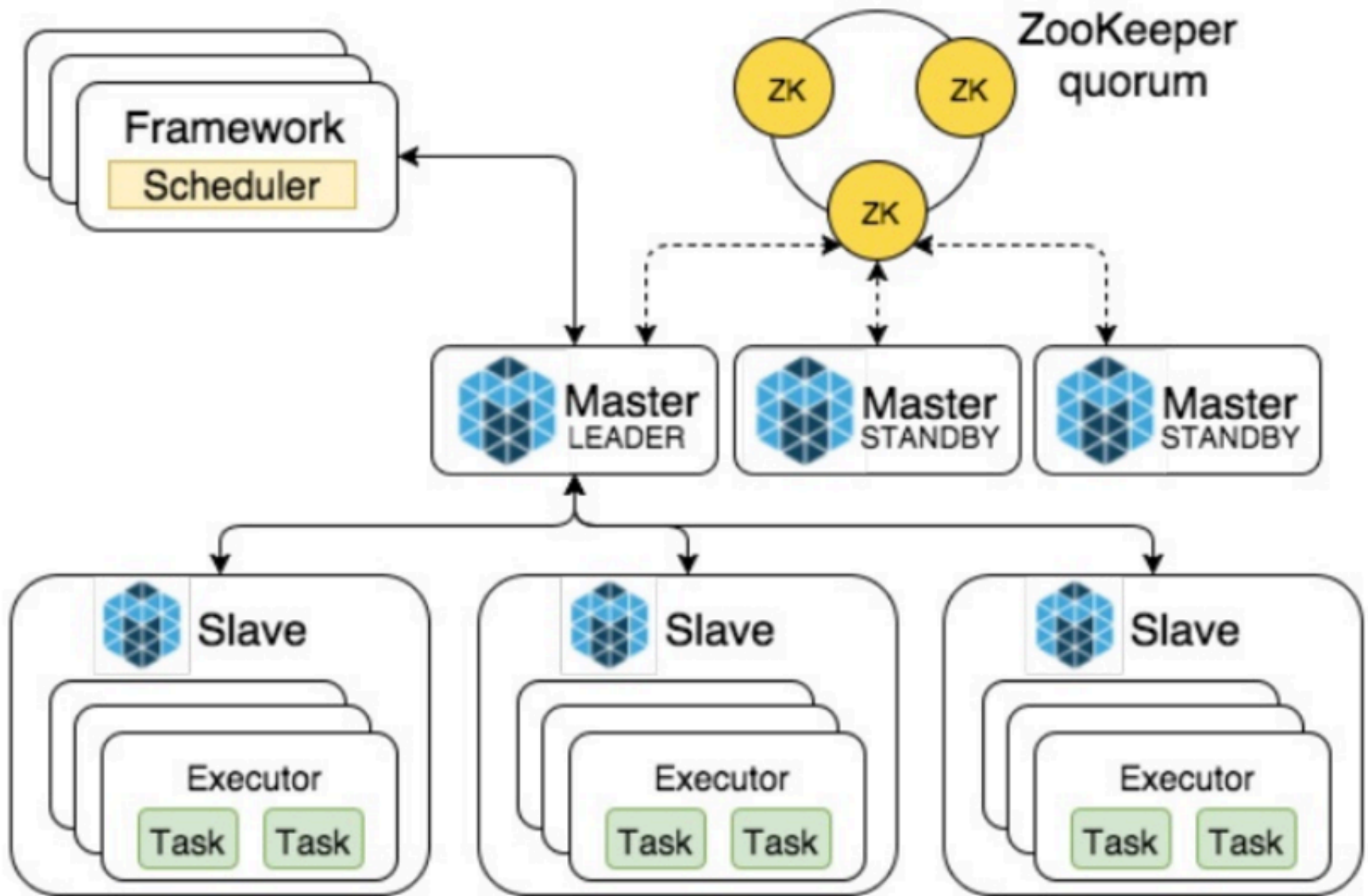
Building data processing platforms with
Spark, **M**esos, **A**kka, **C**assandra and **K**afka



SMACK



Mesos Architecture Overview



Spark Ecosystem

Spark SQL +
DataFrames

Streaming

MLlib
Machine Learning

GraphX
Graph Computation

Spark Core API

R

SQL

Python

Scala

Java

databricks



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.

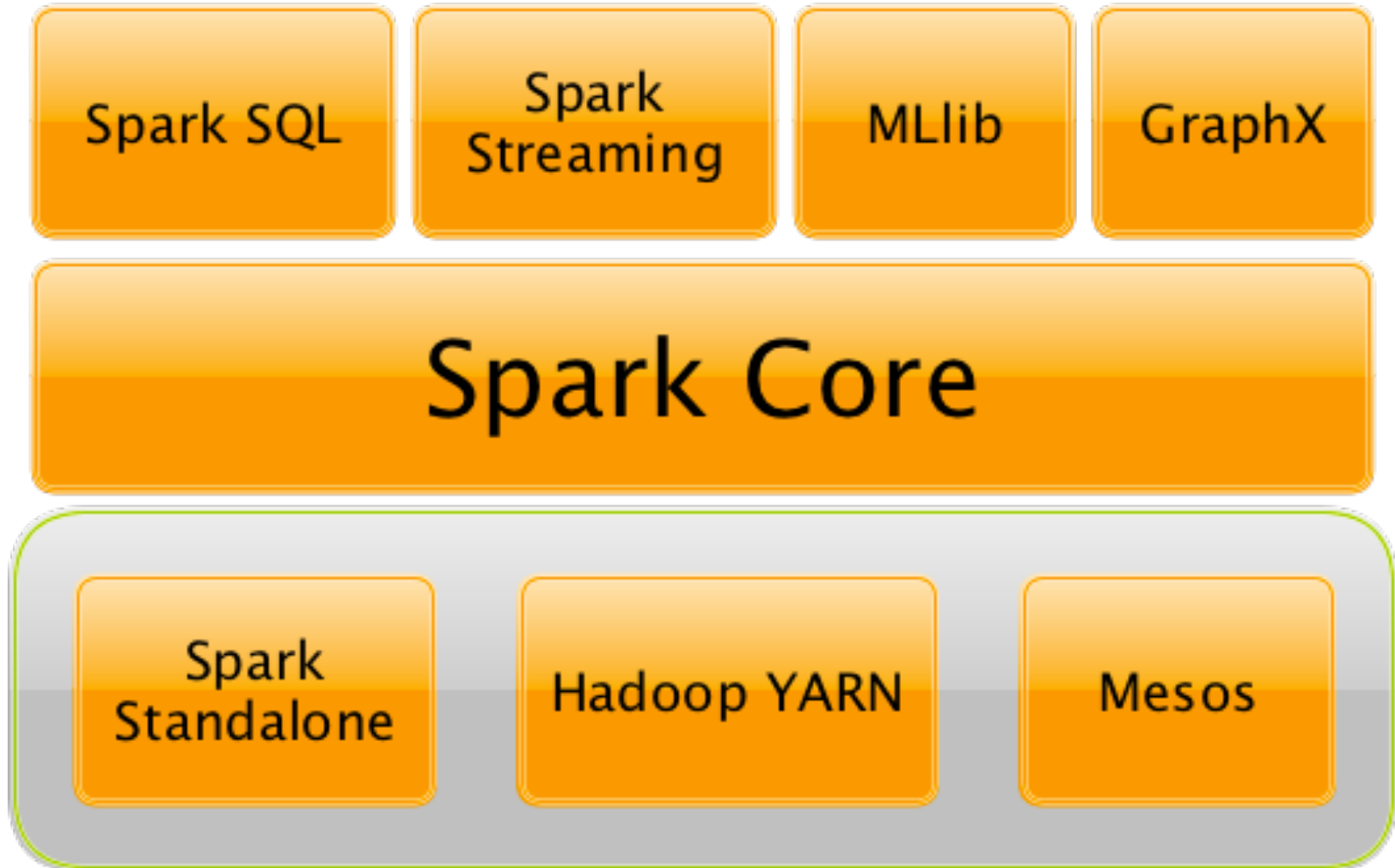
The creators of Spark founded Databricks in 2013.

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

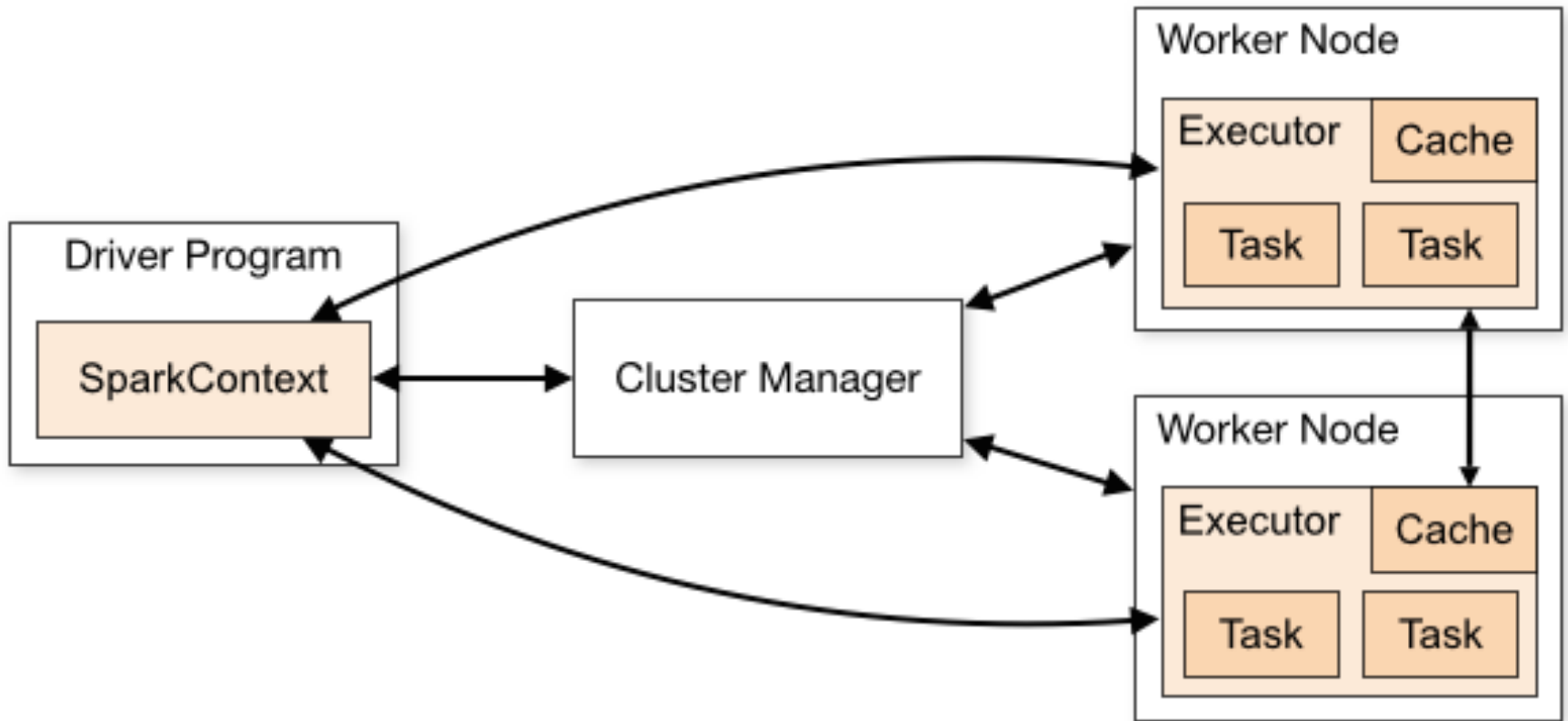
"At Databricks, we're working hard to make Spark easier to use and run than ever, through our efforts on both the Spark codebase and support materials around it. All of our work on Spark is open source and goes directly to Apache."

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

The Spark Platform



Spark Cluster Overview





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.



Running Spark on cluster

Cluster Managers (Schedulers)

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

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:

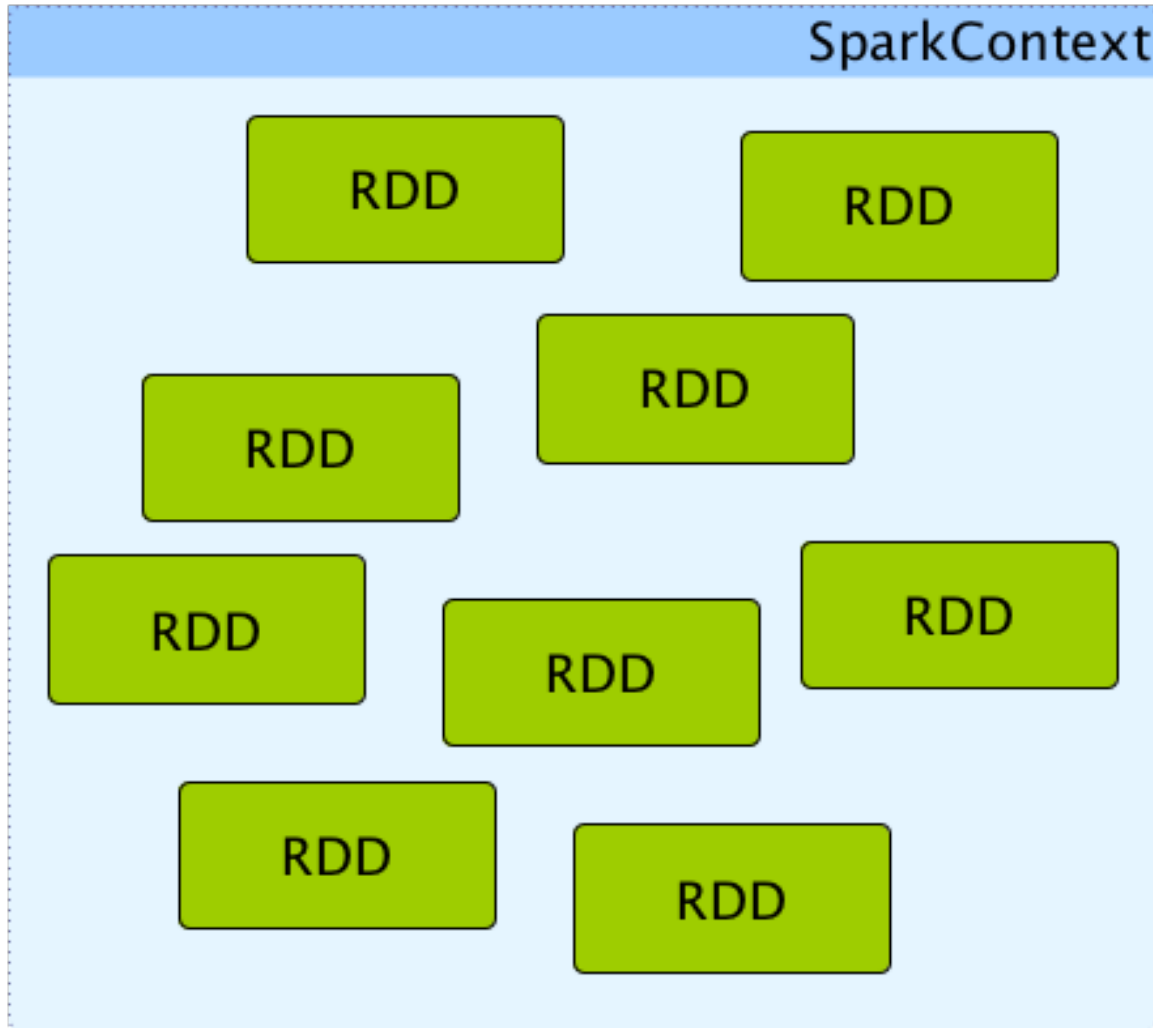
- [Apache Spark channel](#) on YouTube (meetup livestream archives)
- [Spark events worldwide](#)
- [Spark Packages](#)
- [O'Reilly Radar – Spark articles](#)

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](#)

SparkContext and RDDs





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 compu-

tion, 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

RDD

- **Resilient**
 - fault-tolerant and so able to recompute missing or damaged partitions on node failures with the help of **RDD lineage graph**.
- **Distributed** across **clusters**.
- **Dataset**
 - a collection of **partitioned data**.



Resilient Distributed Datasets (RDD)

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

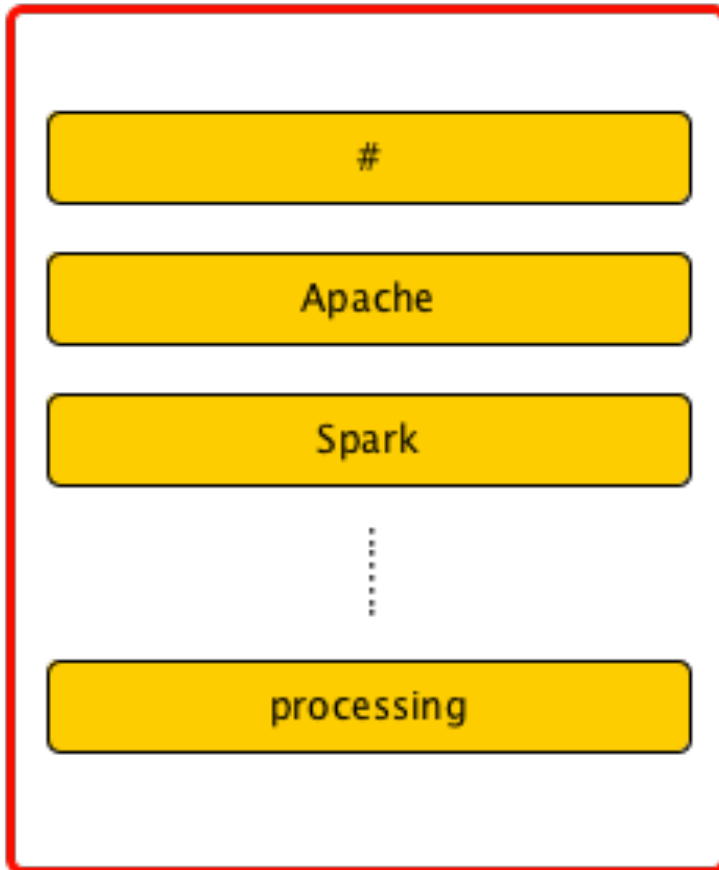


Resilient Distributed Dataset (RDD)

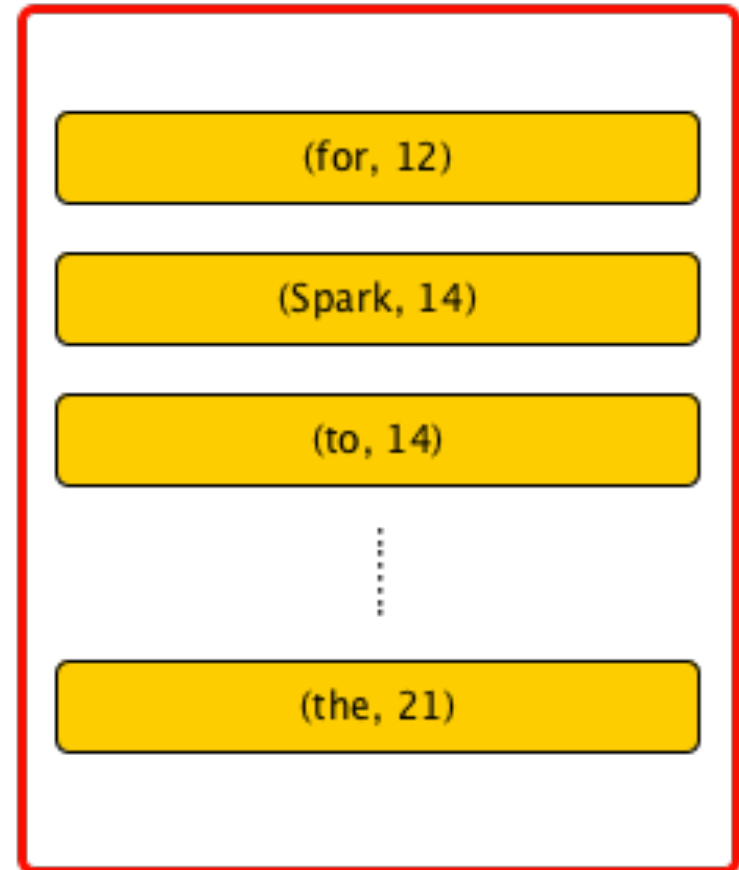
Spark's
“Interface”
to data

Spark RDD

RDD of Strings

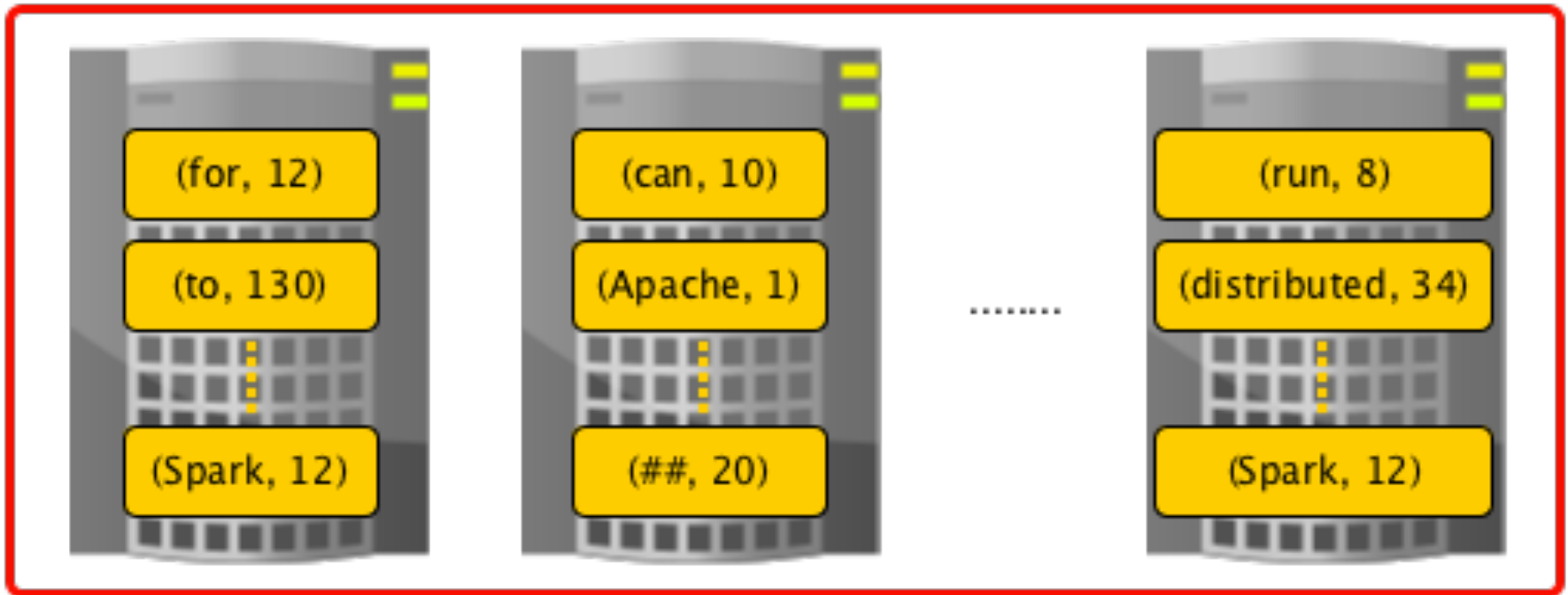


RDD of Pairs

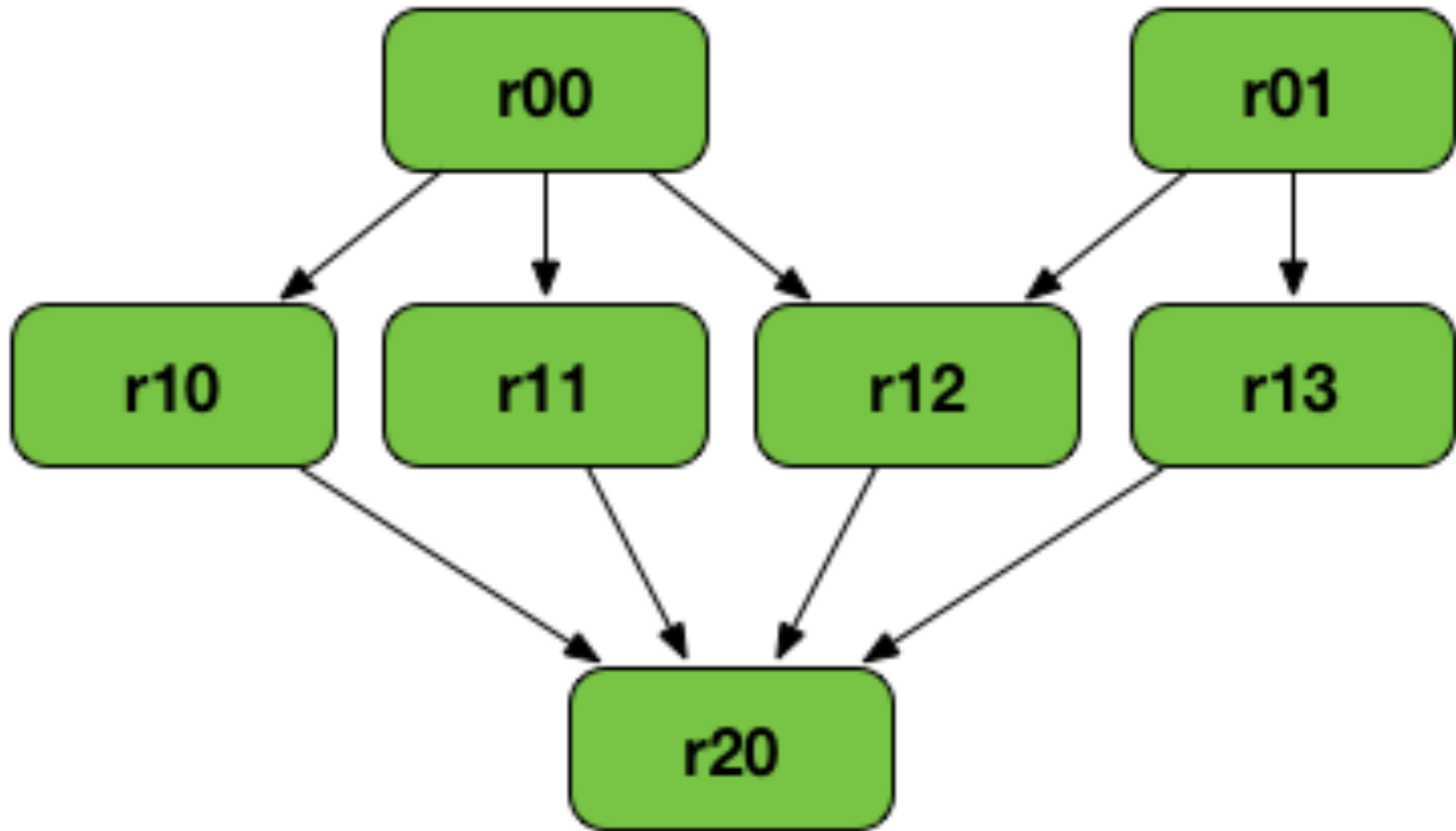


Spark RDD

distributed and partitioned RDD



RDD Lineage Graph (RDD operator graph)

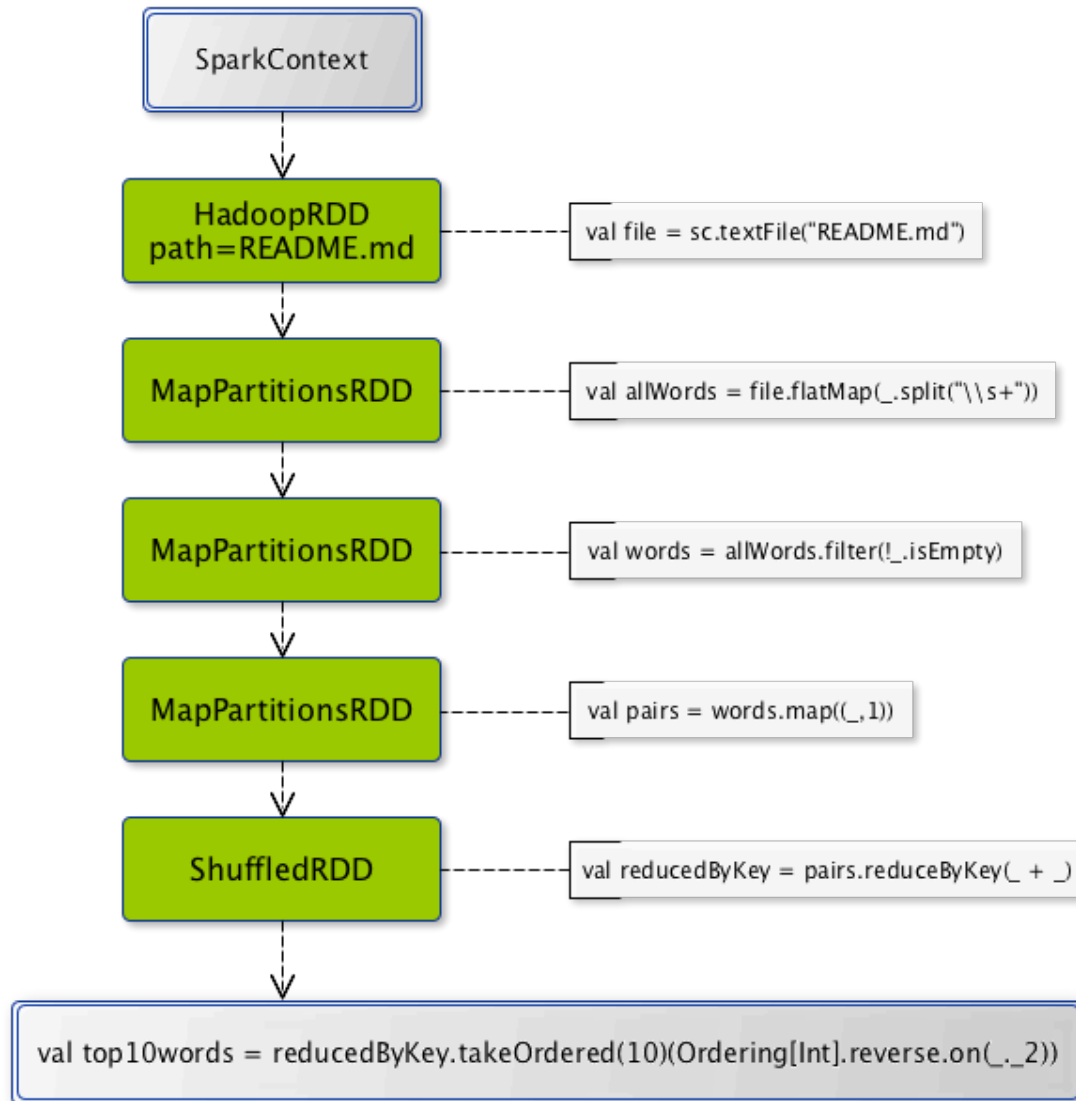


Resilient Distributed Dataset (RDD) Operations

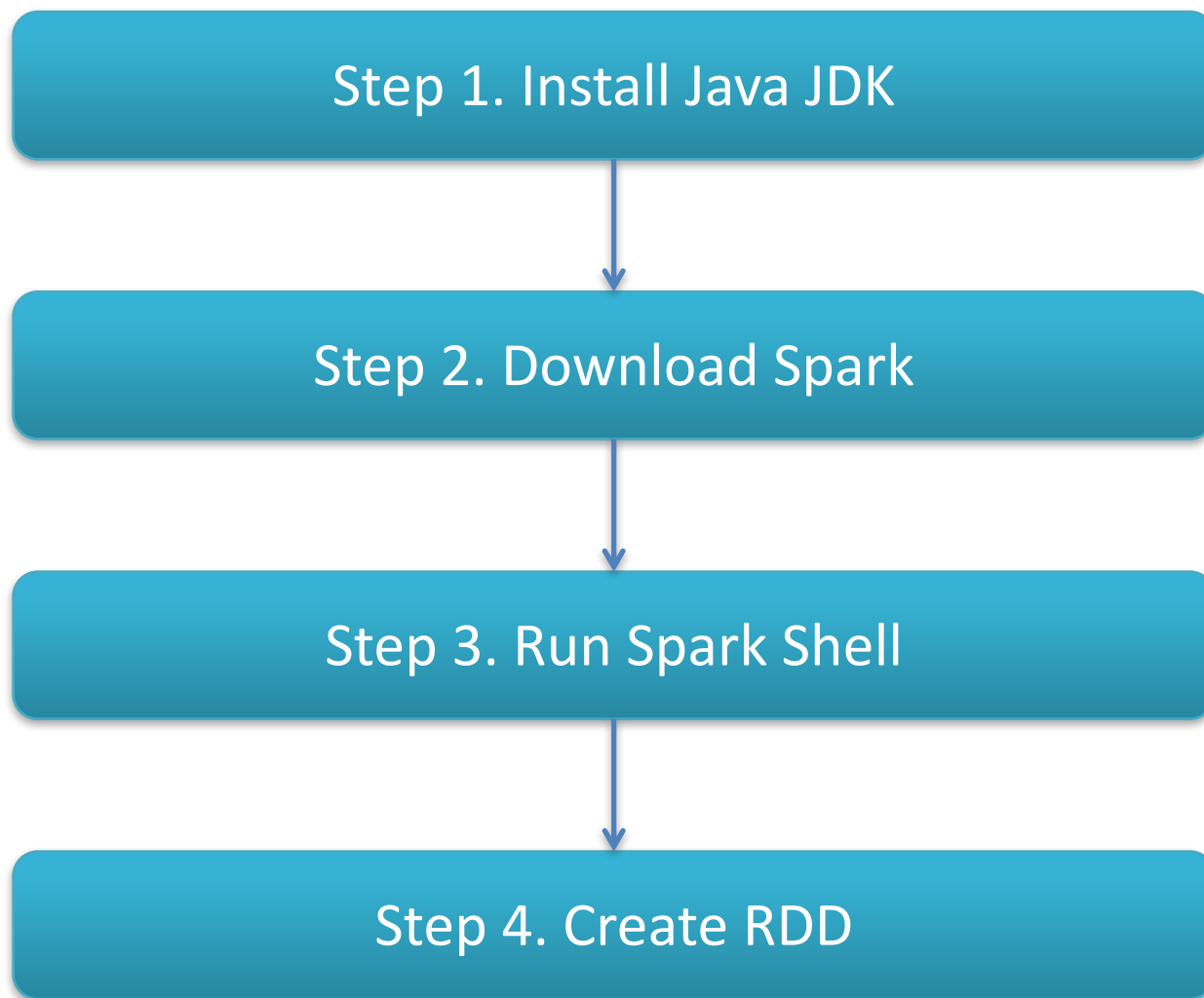
Transformations

Actions

Spark RDD Operations



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



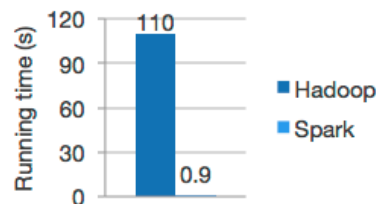
[Download](#)[Libraries ▾](#)[Documentation ▾](#)[Examples](#)[Community ▾](#)[FAQ](#)[Apache Software Foundation ▾](#)

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

Speed

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

Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.



Logistic regression in Hadoop and Spark

Ease of Use

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

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala, Python and R shells.

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

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

Word count in Spark's Python API

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](#)

Spark Download



[Download](#) [Libraries](#) [Documentation](#) [Examples](#) [Community](#) [FAQ](#)

[Apache Software Foundation](#)

Download Apache Spark™

Our latest version is Spark 1.6.0, released on January 4, 2016 ([release notes](#)) ([git tag](#))

1. Choose a Spark release:
2. Choose a package type:
3. Choose a download type:
4. Download Spark: [spark-1.6.0-bin-hadoop2.6.tgz](#)
5. Verify this release using the [1.6.0 signatures and checksums](#).

Note: Scala 2.11 users should download the Spark source package and build [with Scala 2.11 support](#).

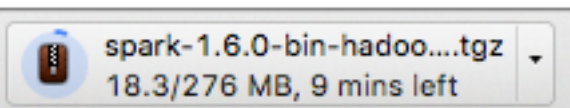
Link with Spark

Spark artifacts are [hosted in Maven Central](#). You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark  
artifactId: spark-core_2.10  
version: 1.6.0
```

Spark Source Code Management

If you are interested in working with the newest under-development code or contributing to Apache Spark development, you can also check out the master branch from Git:



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

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](#)

Scala 2.11


[Overview](#)
[Programming Guides▼](#)
[API Docs▼](#)
[Deploying▼](#)
[More▼](#)

To produce a Spark package compiled with Scala 2.11, use the `-Dscala-2.11` property:

```
./dev/change-scala-version.sh 2.11
mvn -Pyarn -Phadoop-2.4 -Dscala-2.11 -DskipTests clean package
```

Spark does not yet support its JDBC component for Scala 2.11.

```
./dev/change-scala-version.sh 2.11
mvn -Pyarn -Phadoop-2.4 -Dscala-2.11 -DskipTests clean package
```

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

Step 1. Install Java JDK

Spark's interactive shell

```
./bin/spark-shell
```

```
val data = 1 to 10000
```

Step 3. Run Spark Shell

Step 4. Create RDD

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

Step 1. Install Java JDK

```
val distData =  
sc.parallelize(data)  
  
distData.filter(_ < 10).collect()
```

Step 4. Create RDD

Word Count

Hello World
in Big Data

Spark Examples

[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](#)

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://...")
```

Spark Word Count Scala

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

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://...");
```

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://...");
```

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")
```

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>

Scala

[DOCUMENTATION](#)[DOWNLOAD](#)[COMMUNITY](#)[CONTRIBUTE](#)

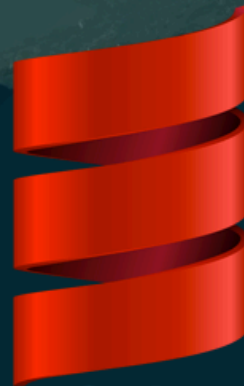
Object-Oriented Meets Functional

Have the best of both worlds. Construct elegant class hierarchies for maximum code reuse and extensibility, implement their behavior using higher-order functions. Or anything in-between.

[LEARN MORE](#)[DOWNLOAD](#)

Getting Started

- Milestones, nightlies, etc.*
- All Previous Releases*



Scala
2.11.7

[API DOCS](#)

Current API Docs

- API Docs (other versions)*
- Scala Documentation*
- Language Specification*

Python

Python

PSF

Docs

PyPI

Jobs

Community



GO

Socialize

Sign In

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```



Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

1

2

3

4

5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Get Started

Download

Docs

Jobs

<https://www.python.org/>

PySpark: Spark Python API



Table Of Contents

Welcome to Spark Python API Docs!

- Core classes:
- Indices and tables

Next topic

pyspark package

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

Welcome to Spark Python API Docs!

Contents:

- pyspark package
 - Subpackages
 - Contents
- pyspark.sql module
 - Module Context
 - pyspark.sql.types module
 - pyspark.sql.functions module
- pyspark.streaming module
 - Module contents
 - pyspark.streaming.kafka module
 - pyspark.streaming.kinesis module
 - pyspark.streaming.flume.module
 - pyspark.streaming.mqtt module
- pyspark.ml package
 - ML Pipeline APIs
 - pyspark.ml.param module
 - pyspark.ml.feature module
 - pyspark.ml.classification module
 - pyspark.ml.clustering module
 - pyspark.ml.recommendation module
 - pyspark.ml.regression module
 - pyspark.ml.tuning module
 - pyspark.ml.evaluation module
- pyspark.mllib package
 - pyspark.mllib.classification module
 - pyspark.mllib.clustering module
 - pyspark.mllib.evaluation module
 - pyspark.mllib.feature module

Anaconda



[Log In](#) [Support](#) [Search](#) [Contact](#)

[ANACONDA](#) [COMMUNITY](#) [CONSULTING](#) [TRAINING](#) [ABOUT](#) [RESOURCES](#)

SUPERPOWERS FOR PEOPLE WHO CHANGE THE WORLD

Leading Open Data Science Platform Powered by Python

DOWNLOAD ANACONDA

ACCELERATE

Time-to-Value



CONNECT

Data, Analytics & Compute



EMPOWER

Data Science Teams



<https://www.continuum.io/>

Anaconda

OPEN DATA SCIENCE PLATFORM

Accelerate. Connect. Empower.



Open Data Science Deep Learning, Machine Learning, AI Interactive Dashboards & Apps
Big Data Data Science Collaboration Data Science Governance Data Science Deployment Self-Service Analytics

OPEN DATA SCIENCE



DATA



COMPUTATION



Anaconda

OPEN DATA SCIENCE PLATFORM

Accelerate. Connect. Empower.



Open Data Science Deep Learning, Machine Learning, AI Interactive Dashboards & Apps

Big Data Data Science Collaboration Data Science Governance Data Science Deployment Self-Service Analytics

OPEN DATA SCIENCE



DATA



COMPUTATION



RESULTS

- ✓ Deploying Data Science models and visualizations easily
- ✓ Building interactive visualizations and web applications
- ✓ Collaboration and sharing of results easily
- ✓ Scaling up & out Data Science to meet performance goals
- ✓ Reproducibility and governance for Data Science assets

Download Anaconda



[Log In](#) [Support](#) [Search](#) [Contact](#)

[ANAconda](#) [COMMUNITY](#) [CONSULTING](#) [TRAINING](#) [ABOUT](#) [RESOURCES](#)

DOWNLOAD ANAconda NOW

Download for



GET SUPERPOWERS WITH ANAconda

Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular Python, R and Scala packages for data science.

Which version should I download and install?

With Anaconda you can run multiple versions of Python in isolated environments, so choose the download with the Python version that you use more often, as that will be your default Python version.

<https://www.continuum.io/downloads>

Download Anaconda Python 3.6

Secure | <https://www.continuum.io/downloads>

Download for Windows

Download for OSX

Download for Linux

Anaconda 4.3.0

For OSX



Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

Graphical Installer

1. Download the graphical installer
2. Double-click the downloaded **.pkg** file and follow the instructions

Command Line Installer

1. Download the command-line installer
2. Optional: Verify data integrity with [MD5](#) or [SHA-256](#) [More info](#)
3. In your terminal window type one of the below and follow the instructions:

Python 3.6 version

```
bash Anaconda3-4.3.0-MacOSX-x86_64.sh
```

Python 3.6 version

GRAPHICAL INSTALLER (424M)

COMMAND-LINE INSTALLER (363M)

64-Bit

Python 2.7 version

GRAPHICAL INSTALLER (419M)

COMMAND-LINE INSTALLER (358M)

64-Bit

<https://www.continuum.io/downloads>

Anadonda for Cloudera CDH



ANACONDA

cloudera

ANACONDA FOR CLOUDERA CDH

Data Science with Python Made Easy for Big Data

GET NOW

Modern Open Source Analytics Powered by Python

Anaconda empowers the entire Data Science team – Data Scientists, Data Engineers, Developers, DevOps, Data Engineers, and Business Analysts – to analyze data in Hadoop and deliver high value, high impact predictive and machine learning solutions with Python.

SEE PACKAGES IN ANACONDA

GET STARTED WITH CLOUDERA
CDH

READ ANACONDA E

✉ Leave us a message

<http://know.continuum.io/anaconda-for-cloudera.html>

Anadonda for Cloudera CDH



Make Python On Hadoop Easy As 1-2-3

- Install Anaconda on a CDH cluster
- Build and run Python based solutions easily across a Cloudera CDH cluster and alongside PySpark jobs

Anadonda for Cloudera CDH



Leverage Anaconda For Distributed Data Science On Cloudera

- Give your Data Scientists the most popular Python packages they know and love
- Empower your Data Science team to explore, build and deploy predictive models

Anadonda for Cloudera CDH



Transition To Enterprise To Get Full Power Of Anaconda

- Dynamically manage multiple Python environments alongside a CDH cluster
- Get security, authentication and many other enterprise requirements with the paid Anaconda Workgroup and Anaconda Enterprise subscriptions
- Read our blog on [Getting the Most Out of Anaconda for Your Cluster](http://know.continuum.io/anaconda-for-cloudera.html).

Download Cloudera Enterprise

Local, On Premise, or Cloud-based Apache Hadoop Management



QuickStarts

Get Started on your local machine using a QuickStart VM or Docker Image.

[DOWNLOAD NOW](#)

[Learn More](#)



Cloudera Manager

A unified interface to manage your enterprise data hub. Express and Enterprise editions available.

[DOWNLOAD NOW](#)

[Learn More](#)



Cloudera Director

Self-service, reliable experience for CDH and Cloudera Enterprise in the cloud

[DOWNLOAD NOW](#)

[Learn More](#)

Cloudera CDH 5.5

[Downloads](#) [Training](#) [Support Portal](#) [Partners](#) [Developers](#) [Community](#)

[Search](#) [Sign In](#) [Language](#)

cloudera

[Why Cloudera](#) [Products](#) [Services & Support](#) [Solutions](#) [Get Started](#)

QuickStart Downloads for CDH 5.5

Easy-to-deploy Apache Hadoop clusters for easy learning!

Cloudera QuickStart downloads contain complete Apache Hadoop clusters in the form of VMs or Docker images, including Cloudera Manager to manage them.

Cloudera QuickStart downloads are for personal and demo purposes only, and are not to be used as a starting point for production clusters.

Get Started

QUICKSTART DOWNLOADS FOR CDH 5.5 ▾

SELECT A PLATFORM ▾

Docker Image

KVM

Virtual Box

VMWare

System Requirements

Installed Products

Getting Started

http://www.cloudera.com/downloads/quickstart_vms/5-5.html

Cloudera CDH 5.5

[Downloads](#) [Training](#) [Support Portal](#) [Partners](#) [Developers](#) [Community](#)

[Search](#) [Sign In](#) [Language](#)

cloudera

[Why Cloudera](#) [Products](#) [Services & Support](#) [Solutions](#) [Get Started](#)

QuickStart Downloads for CDH 5.5

Easy-to-deploy Apache Hadoop clusters for easy learning!

Cloudera QuickStart downloads contain complete Apache Hadoop clusters in the form of VMs or Docker images, including Cloudera Manager to manage them.

Cloudera QuickStart downloads are for personal and demo purposes only, and are not to be used as a starting point for production clusters.

Get Started

QUICKSTART DOWNLOADS FOR CDH 5.5 ▾

VIRTUAL BOX ▾

DOWNLOAD NOW 

System Requirements

Installed Products

Getting Started

http://www.cloudera.com/downloads/quickstart_vms/5-5.html

Cloudera CDH

[Downloads](#) [Training](#) [Support Portal](#) [Partners](#) [Developers](#) [Community](#)

[Search](#) [Sign In](#) [Language](#)

cloudera

[Why Cloudera](#) [Products](#) [Services & Support](#) [Solutions](#) [Get Started](#)

CDH Components

CDH is Cloudera's 100% open source platform distribution, including Apache Hadoop and built specifically to meet enterprise demands. CDH delivers everything you need for enterprise use right out of the box. By integrating Hadoop with more than a dozen other critical open source projects, Cloudera has created a functionally advanced system that helps you perform end-to-end Big Data workflows.

[TRY CDH](#)

Spark Programming Guide

- Overview
- Linking with Spark
- Initializing Spark
 - Using the Shell
- Resilient Distributed Datasets (RDDs)
 - Parallelized Collections
 - External Datasets
 - RDD Operations
 - Basics
 - Passing Functions to Spark
 - Understanding closures
 - Example
 - Local vs. cluster modes
 - Printing elements of an RDD
 - Working with Key-Value Pairs
 - Transformations
 - Actions
 - Shuffle operations
 - Background
 - Performance Impact
 - RDD Persistence
 - Which Storage Level to Choose?

References

- Mike Frampton (2015), Mastering Apache Spark, Packt Publishing
- Anton Kirillov (2015), Data processing platforms architectures with Spark, Mesos, Akka, Cassandra and Kafka, Big Data AW Meetup, <http://www.slideshare.net/akirillov/data-processing-platforms-architectures-with-spark-mesos-akka-cassandra-and-kafka>
- Spark Apache (2016), PySpark: Spark Python API, <http://spark.apache.org/docs/latest/api/python/>
- Spark Apache (2016), Spark Programming Guide, <http://spark.apache.org/docs/latest/programming-guide.html>
- Spark Submit (2014), Spark Summit 2014 Training, <https://spark-summit.org/2014/training>
- Databricks (2016), Spark Developer Resources, <https://databricks.com/spark/developer-resources>
- Liondatasystems (2015), Intro to Apache Spark (Brain-Friendly Tutorial), <https://www.youtube.com/watch?v=rvDpBTV89AM>
- Databricks (2016), Intro to Apache Spark, http://training.databricks.com/workshop/itas_workshop.pdf