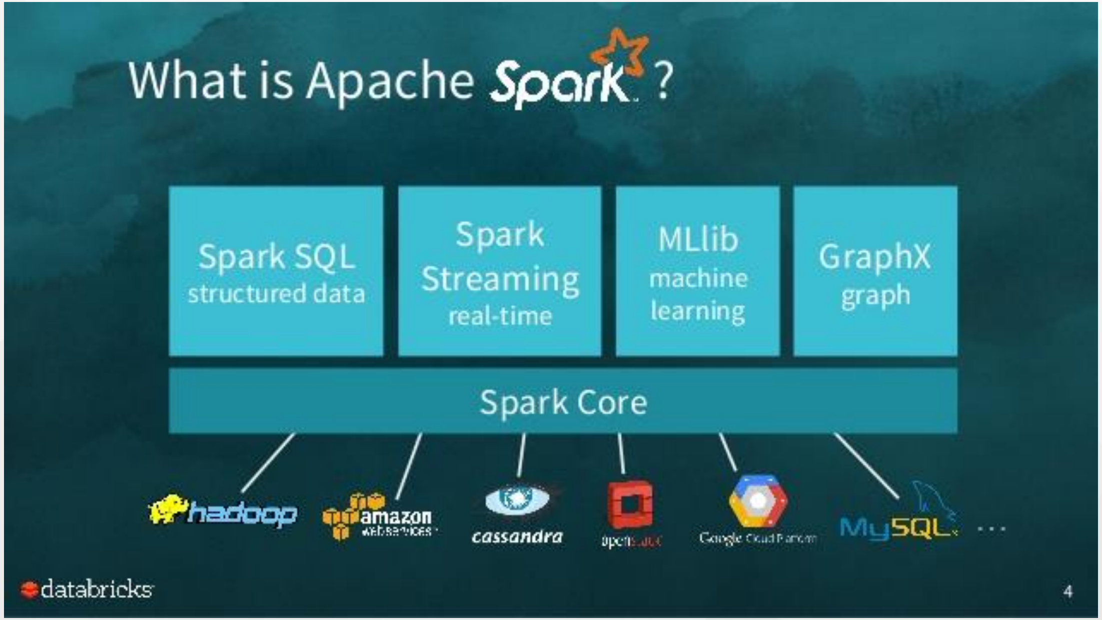


What is Apache Spark ?

Spark SQL
structured data

Spark Streaming
real-time

MLlib
machine learning

GraphX
graph

Spark Core

The Pilot Of Spark

2017.5 XenRon

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

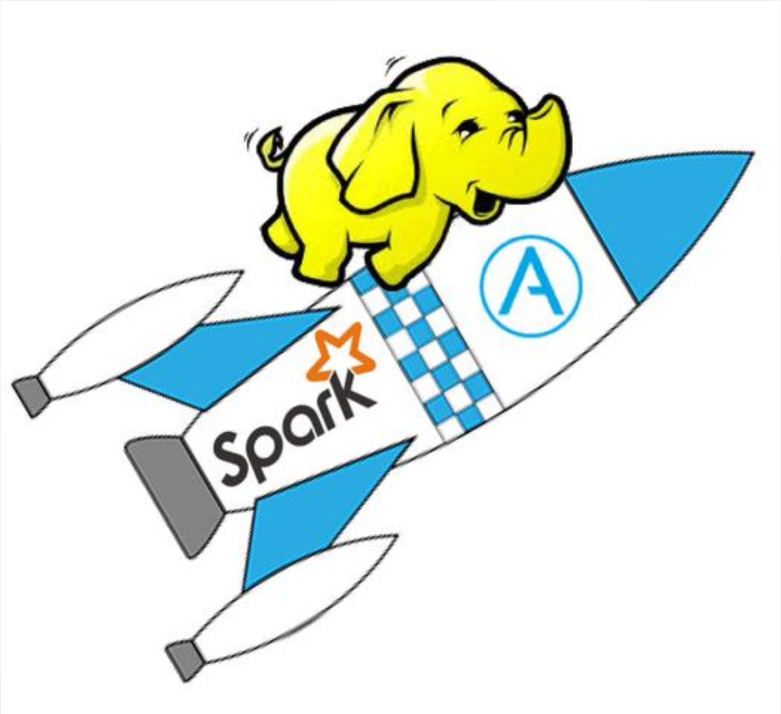
CONTENTS

Preliminary Topics 01

Spark Environment 02

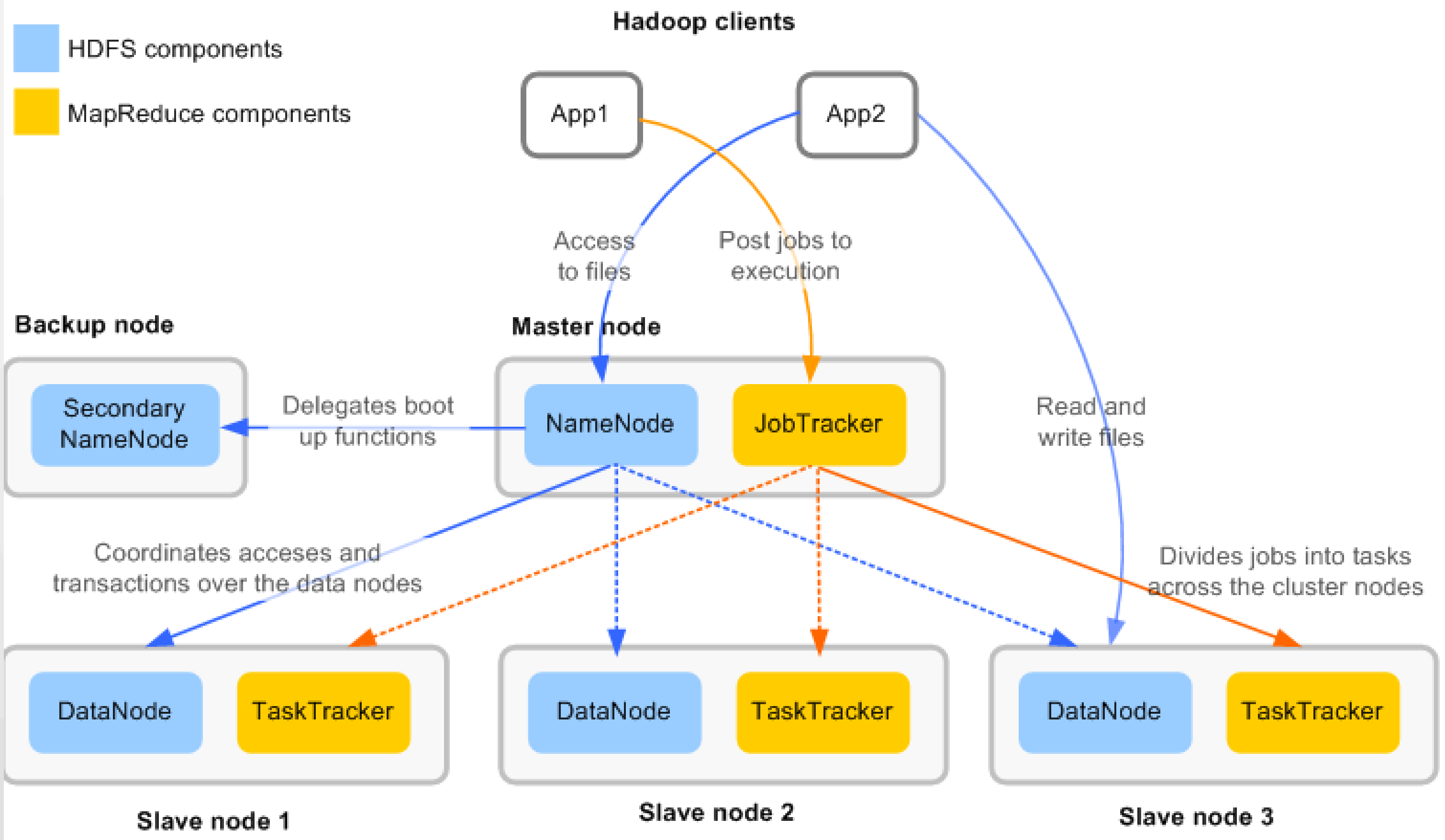
Spark Architecture 03

Spark EcoSystem 04



Review

Map Reduce


HDFS components
MapReduce components
Hadoop clients
App1
App2
Access to files
Post jobs to execution
Backup node
Master node
Secondary NameNode
Delegates boot up functions
NameNode
JobTracker
Read and write files
Coordinates acceses and transactions over the data nodes
Divides jobs into tasks across the cluster nodes
DataNode
TaskTracker
DataNode
TaskTracker
DataNode
TaskTracker
Slave node 1
Slave node 2
Slave node 3

HDFS

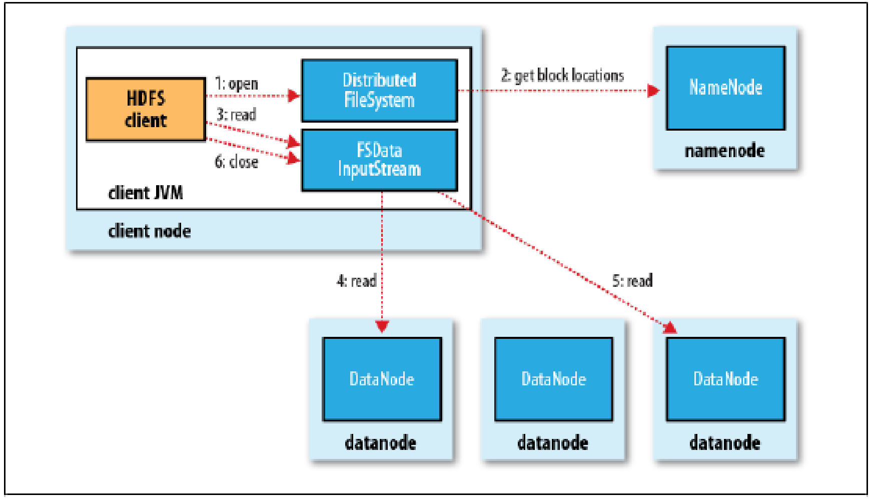


Figure 3-2. A client reading data from HDFS

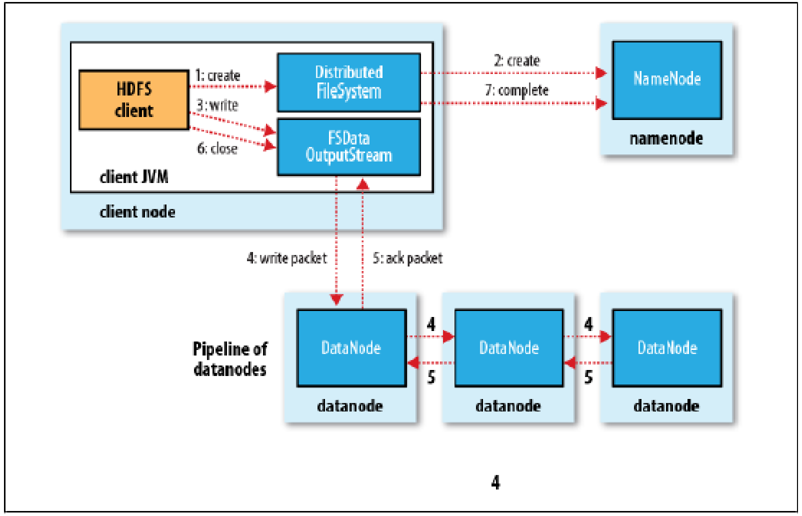
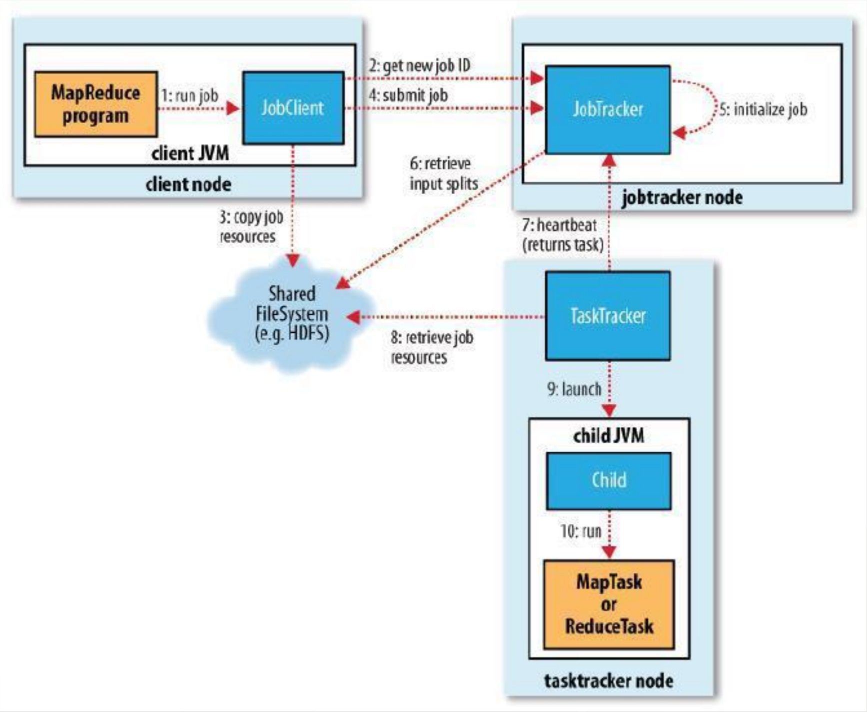


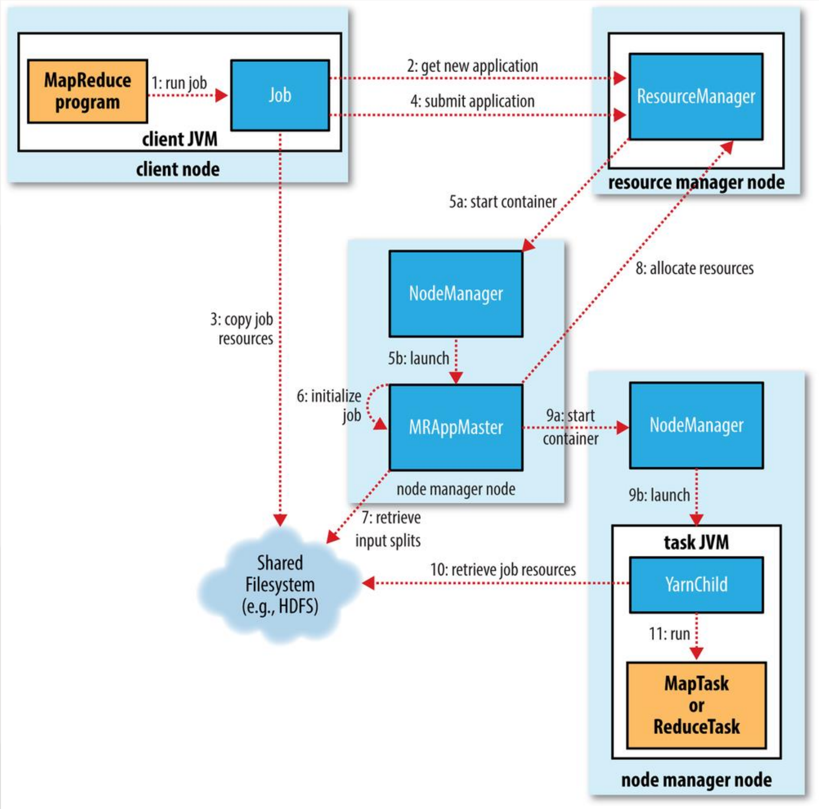
Figure 3-4. A client writing data to HDFS

Map Reduce

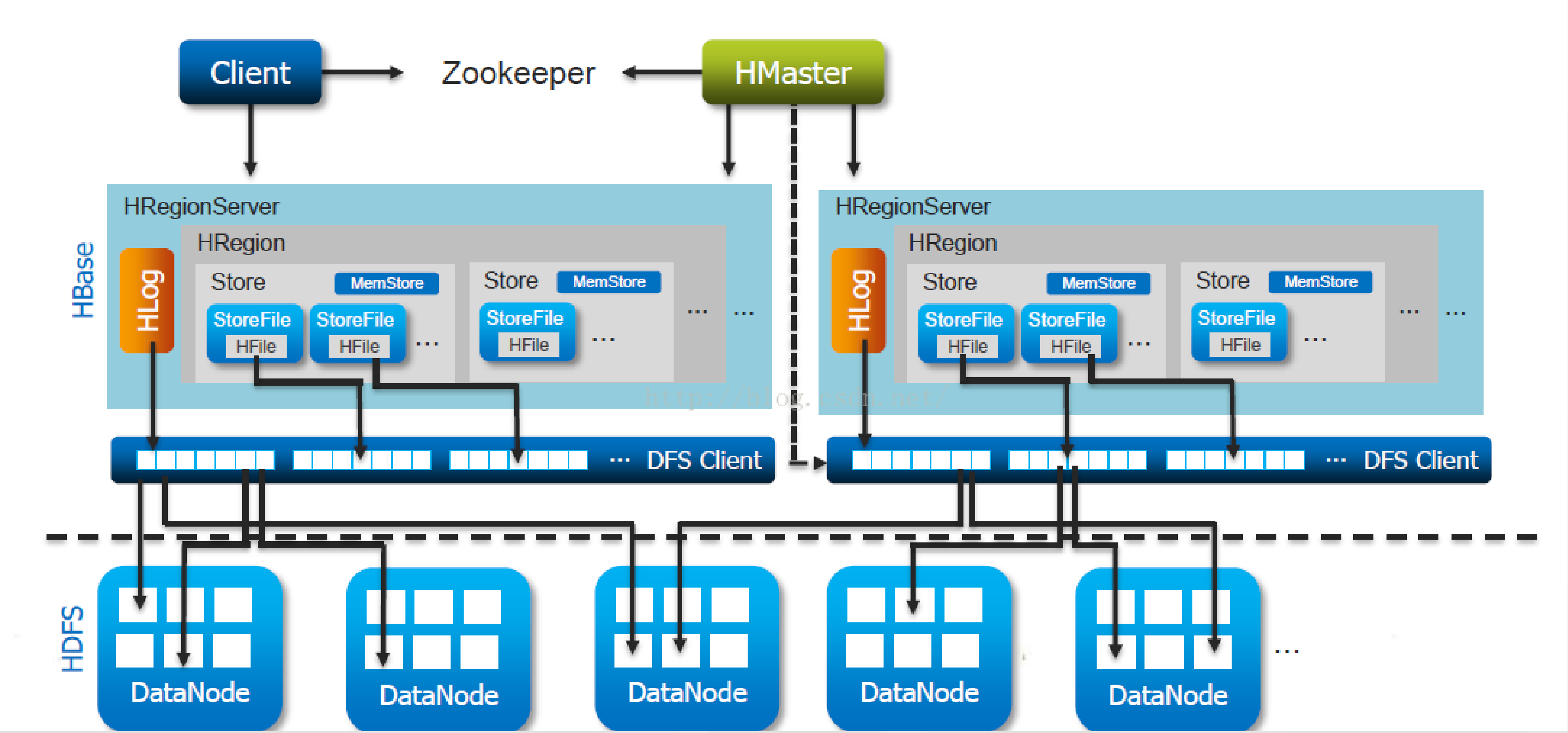
Hadoop 1.x



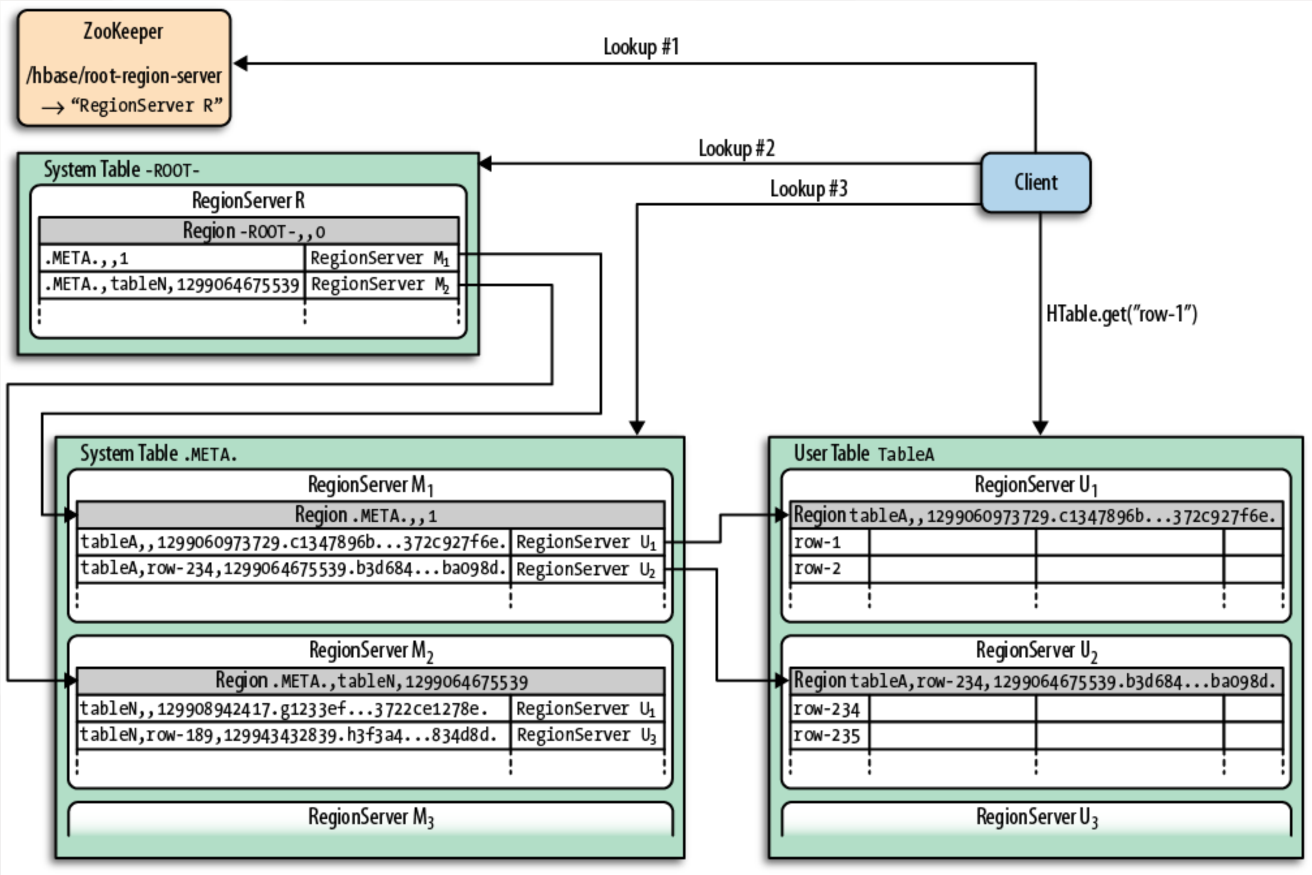
Hadoop 2.x



HBase


Client
Zookeeper
HMaster
HRegionServer
HRegion
HLog
Store
MemStore
StoreFile
HFile
...
DFS Client
HRegionServer
HRegion
HLog
Store
MemStore
StoreFile
HFile
...
DFS Client
HBase
HDFS
DataNode
DataNode
DataNode
DataNode
DataNode
...

HBase



System Table -ROOT-

RegionServer R

Region -ROOT-,,0	
.META.,,1	RegionServer M_1
.META.,tableN,1299064675539	RegionServer M_2

System Table .META.

RegionServer M_1

Region .META.,,1	
tableA,,1299060973729.c1347896b...372c927f6e.	RegionServer U_1
tableA,row-234,1299064675539.b3d684...ba098d.	RegionServer U_2

RegionServer M_2

Region .META.,tableN,1299064675539	
tableN,,129908942417.g1233ef...3722ce1278e.	RegionServer U_1
tableN,row-189,129943432839.h3f3a4...834d8d.	RegionServer U_3

RegionServer M_3

User Table TableA

RegionServer U_1

Region tableA,,1299060973729.c1347896b...372c927f6e.			
row-1			
row-2			

RegionServer U_2

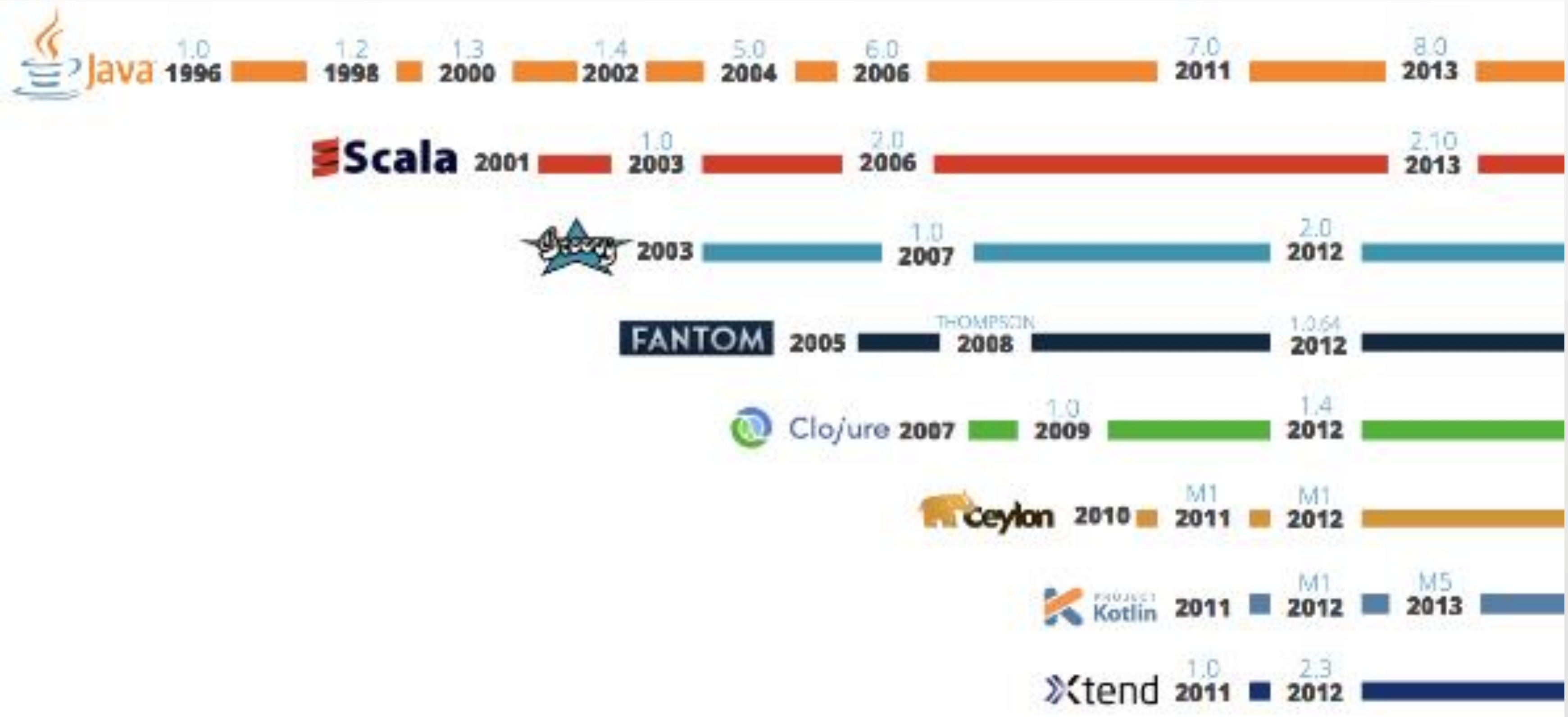
Region tableA,row-234,1299064675539.b3d684...ba098d.			
row-234			
row-235			

RegionServer U_3

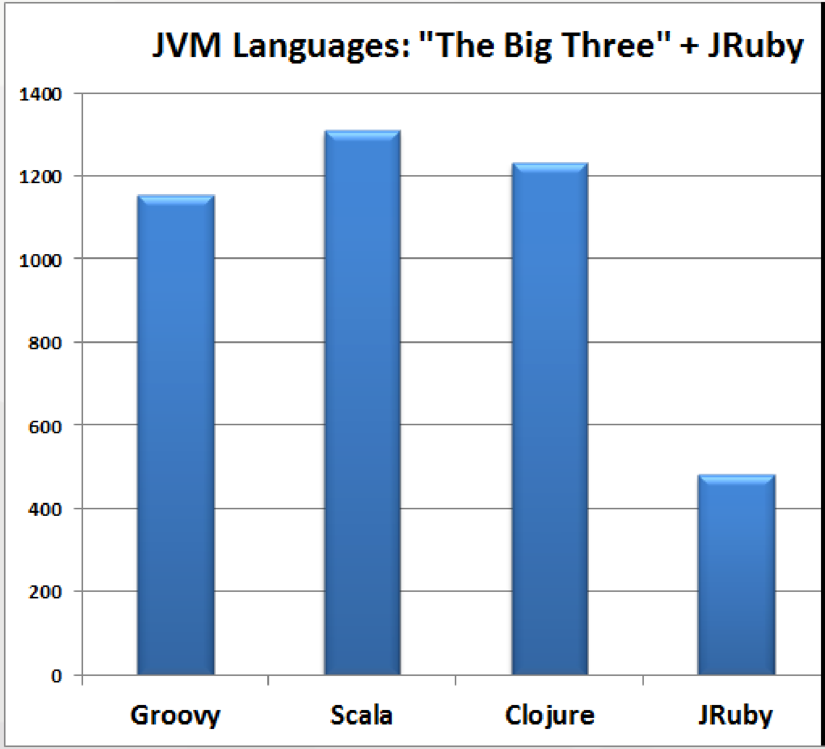
PART1

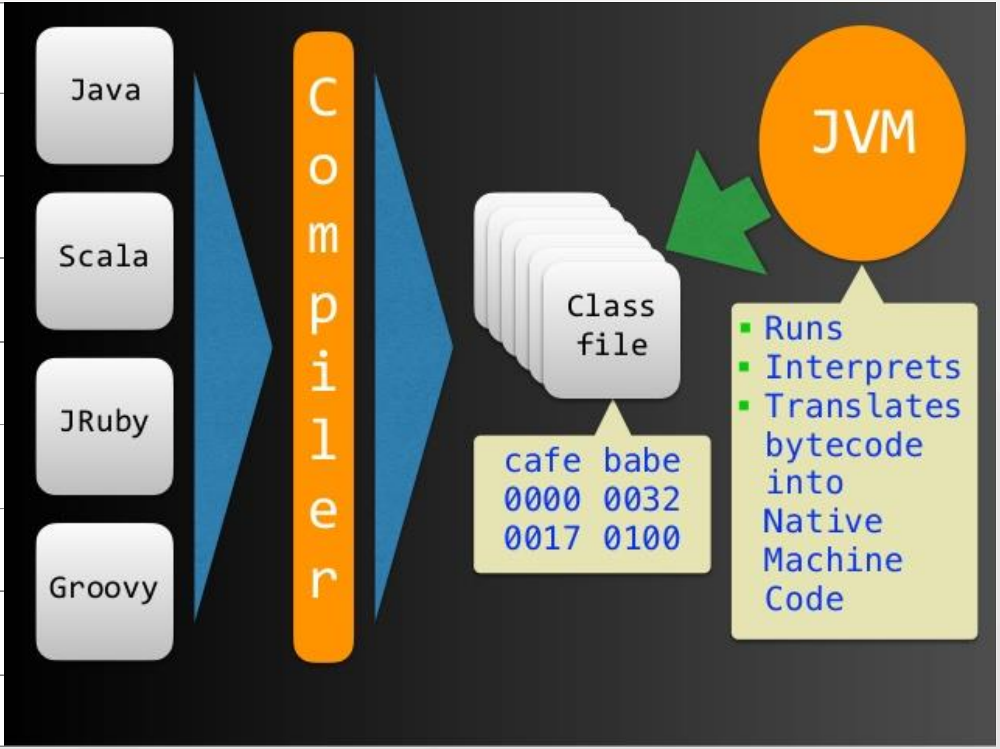
Preliminary Topics
事前準備

JVM


Java
1.0 1996
1.2 1998
1.3 2000
1.4 2002
5.0 2004
6.0 2006
7.0 2011
8.0 2013
Scala
2001
1.0 2003
2.0 2006
2.10 2013
Groovy
2003
1.0 2007
2.0 2012
FANTOM
2005
THOMPSON 2008
1.0.64 2012
Clojure
2007
1.0 2009
1.4 2012
Ceylon
2010
M1 2011
M1 2012
PROJECT Kotlin
2011
M1 2012
M5 2013
Xtend
1.0 2011
2.3 2012

Big Three




Java

Scala

JRuby

Groovy

Compiler

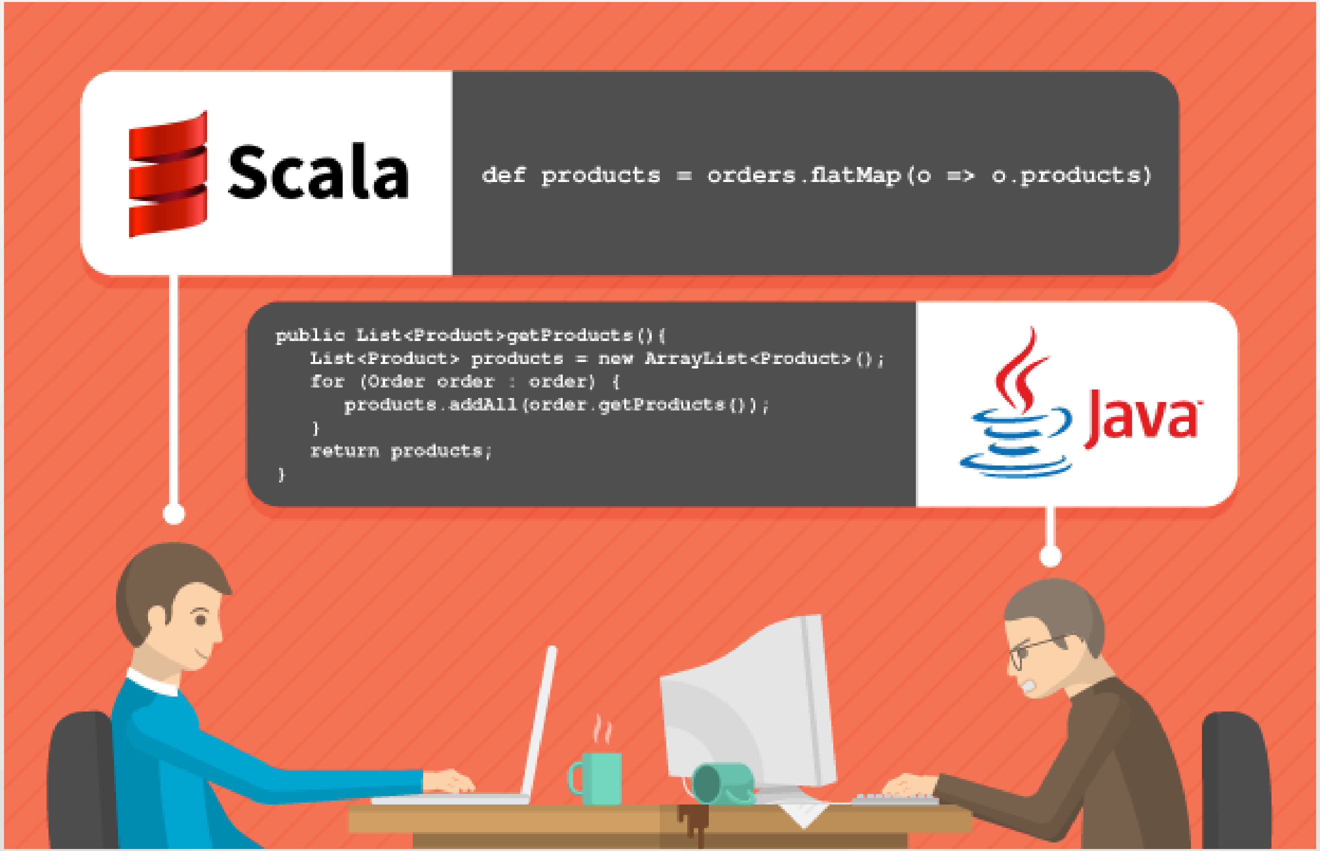
Class file

```
cafe babe
0000 0032
0017 0100
```

JVM

- Runs
- Interprets
- Translates bytecode into Native Machine Code

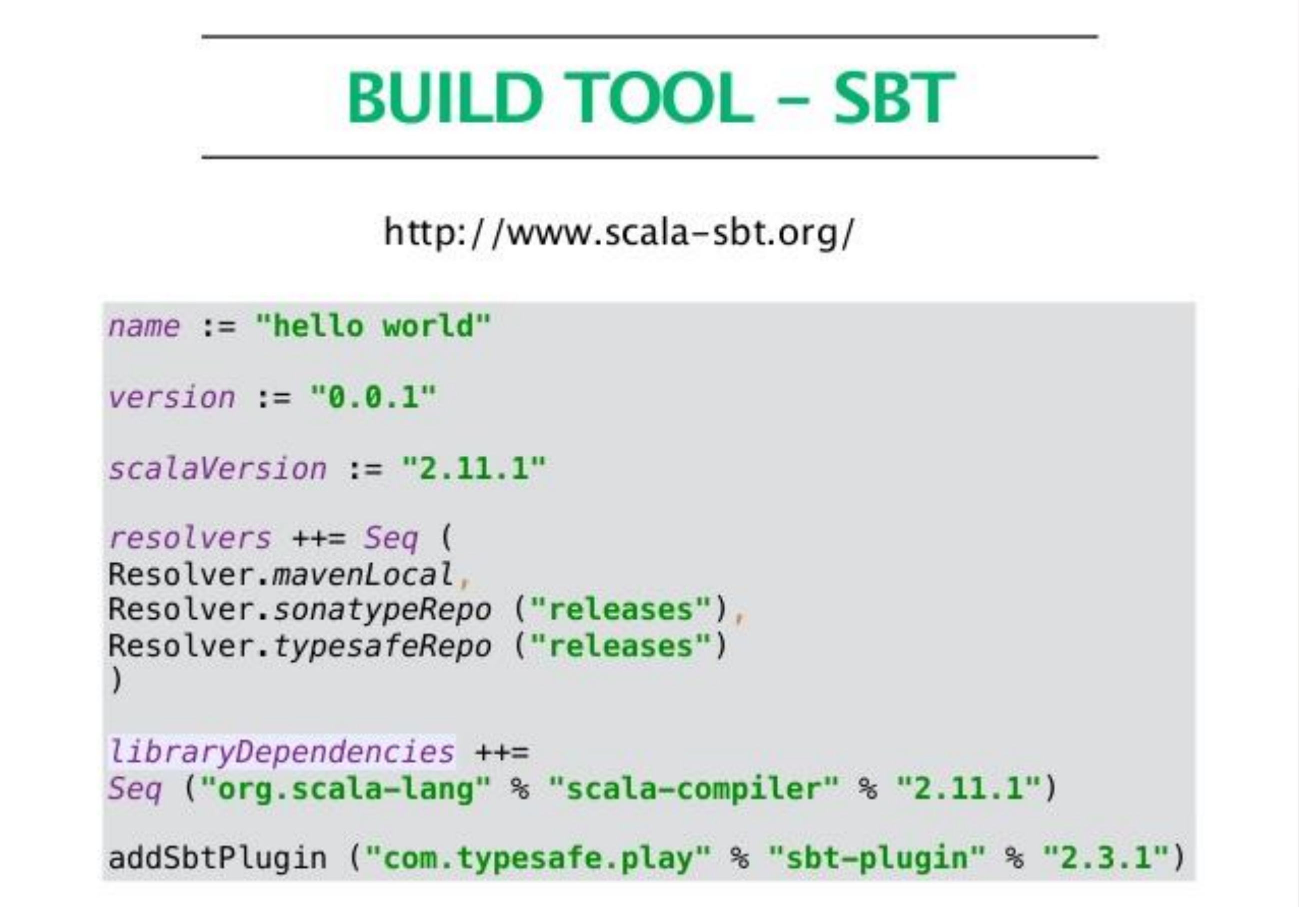
Scala



```
def products = orders.flatMap(o => o.products)
```

```
public List<Product>getProducts(){
   List<Product> products = new ArrayList<Product>();
   for (Order order : order) {
      products.addAll(order.getProducts());
   }
   return products;
}
```

Sbt

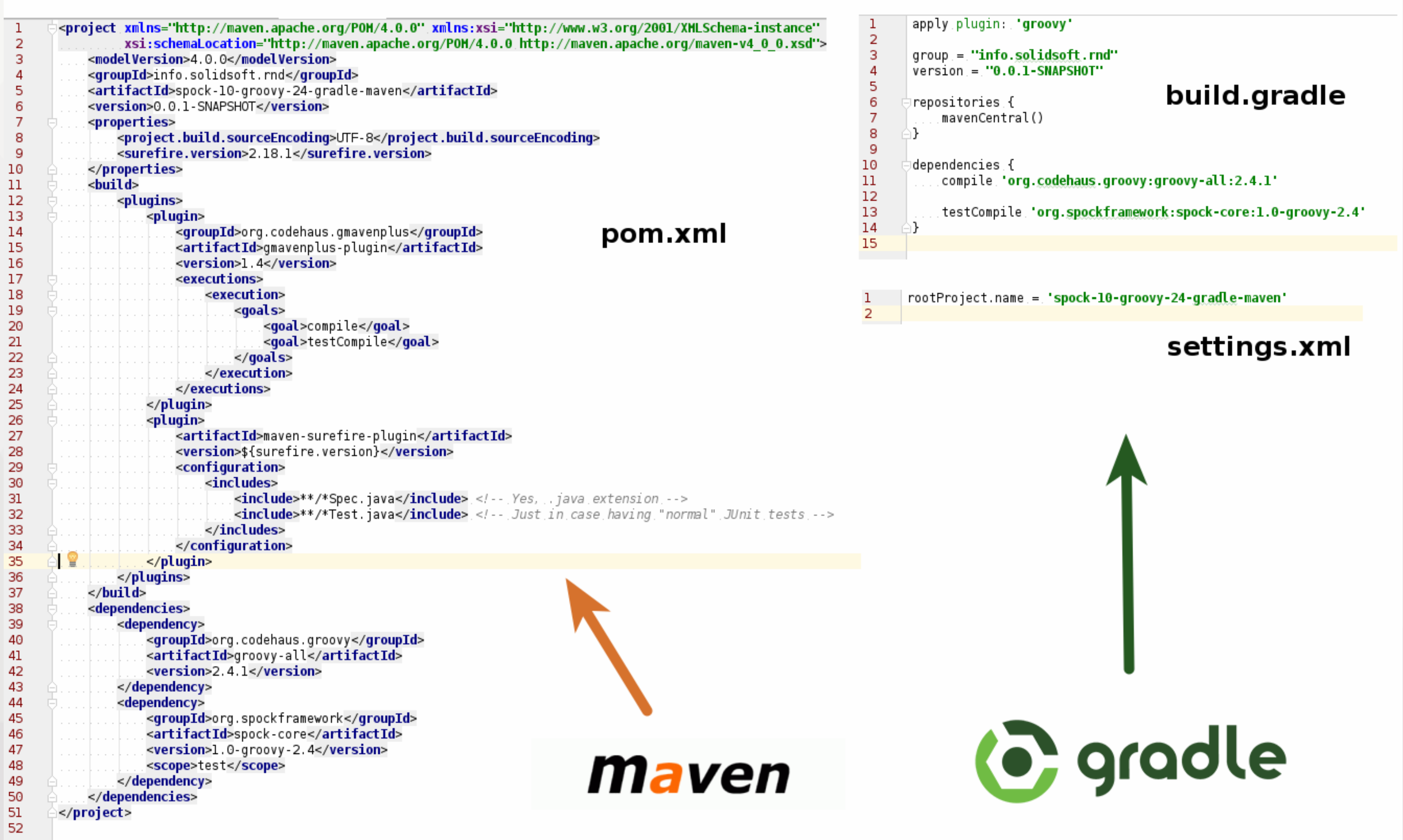


BUILD TOOL – SBT

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

```
name := "hello world"

version := "0.0.1"

scalaVersion := "2.11.1"

resolvers ++= Seq (
Resolver.mavenLocal,
Resolver.sonatypeRepo ("releases"),
Resolver.typesafeRepo ("releases")
)

libraryDependencies ++=
Seq ("org.scala-lang" % "scala-compiler" % "2.11.1")

addSbtPlugin ("com.typesafe.play" % "sbt-plugin" % "2.3.1")
```

Sbt



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>info.solidsoft.rnd</groupId>
    <artifactId>spock-10-groovy-24-gradle-maven</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <surefire.version>2.18.1</surefire.version>
    </properties>
    <build>
        <plugins>
            <plugin>
                <groupId>org.codehaus.gmavenplus</groupId>
                <artifactId>gmavenplus-plugin</artifactId>
                <version>1.4</version>
                <executions>
                    <execution>
                        <goals>
                            <goal>compile</goal>
                            <goal>testCompile</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
            <plugin>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>${surefire.version}</version>
                <configuration>
                    <includes>
                        <include>**/*Spec.java</include> <!-- Yes, .java extension -->
                        <include>**/*Test.java</include> <!-- Just in case having "normal" JUnit tests -->
                    </includes>
                </configuration>
            </plugin>
        </plugins>
    </build>
    <dependencies>
        <dependency>
            <groupId>org.codehaus.groovy</groupId>
            <artifactId>groovy-all</artifactId>
            <version>2.4.1</version>
        </dependency>
        <dependency>
            <groupId>org.spockframework</groupId>
            <artifactId>spock-core</artifactId>
            <version>1.0-groovy-2.4</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>

```

pom.xml

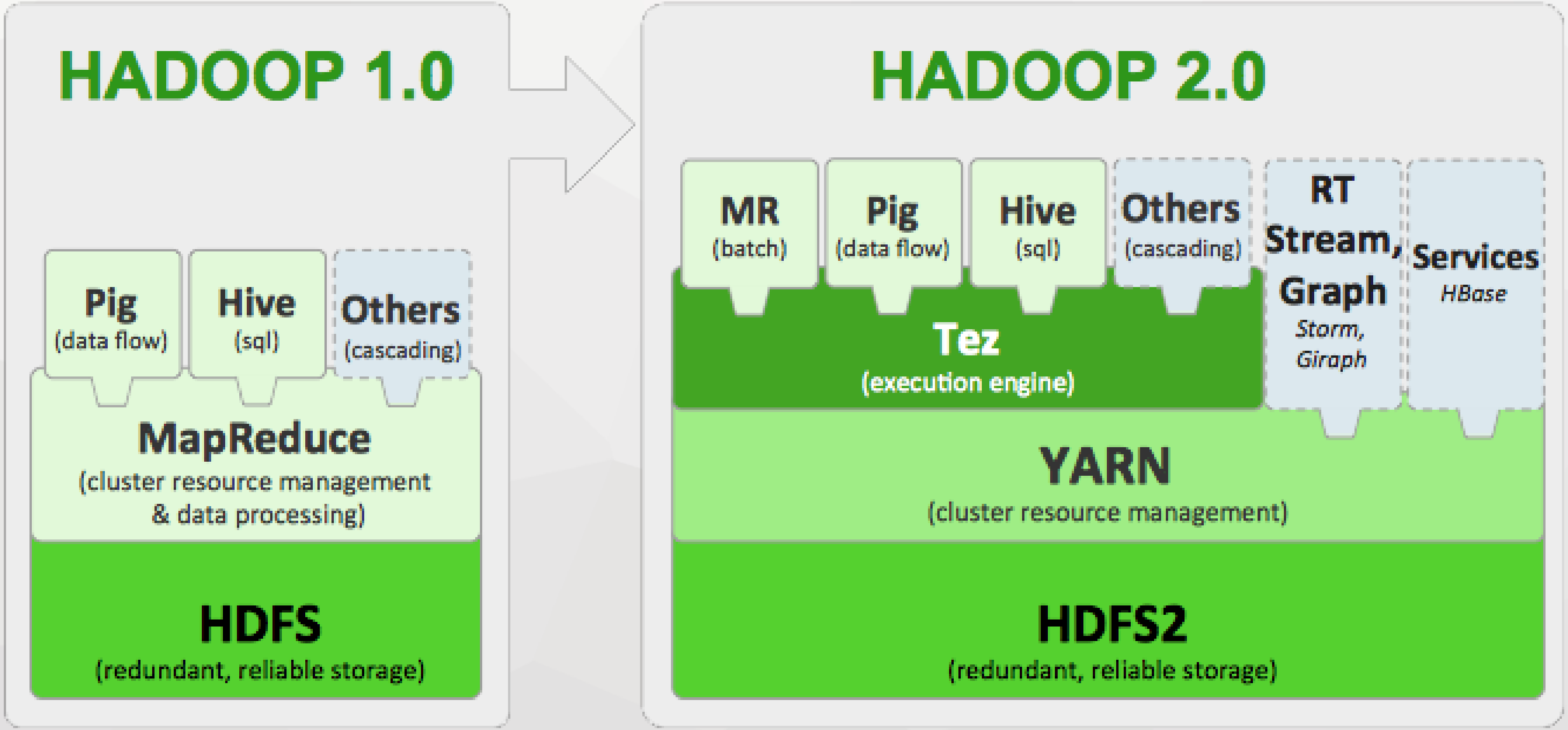
```groovy
apply plugin: 'groovy'

group = "info.solidsoft.rnd"
version = "0.0.1-SNAPSHOT"

repositories {
    mavenCentral()
}

dependencies {
    compile 'org.codehaus.groovy:groovy-all:2.4.1'

    testCompile 'org.spockframework:spock-core:1.0-groovy-2.4'
}

```

build.gradle

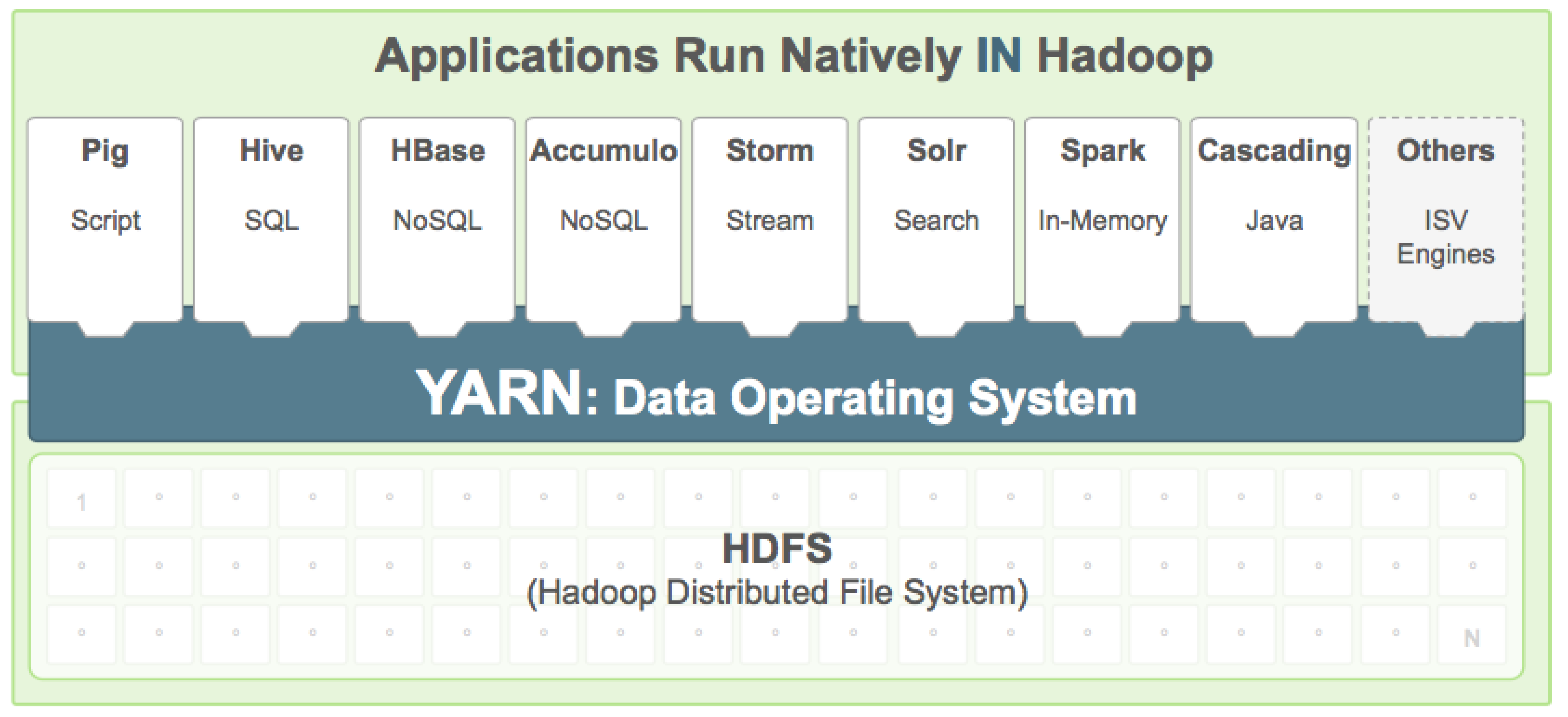
```groovy
rootProject.name = 'spock-10-groovy-24-gradle-maven'

```

settings.xml

Yarn


HADOOP 1.0
Pig
(data flow)
Hive
(sql)
Others
(cascading)
MapReduce
(cluster resource management
& data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MR
(batch)
Pig
(data flow)
Hive
(sql)
Others
(cascading)
RT
Stream,
Graph
Storm,
Giraph
Services
HBase
Tez
(execution engine)
YARN
(cluster resource management)
HDFS2
(redundant, reliable storage)

Yarn



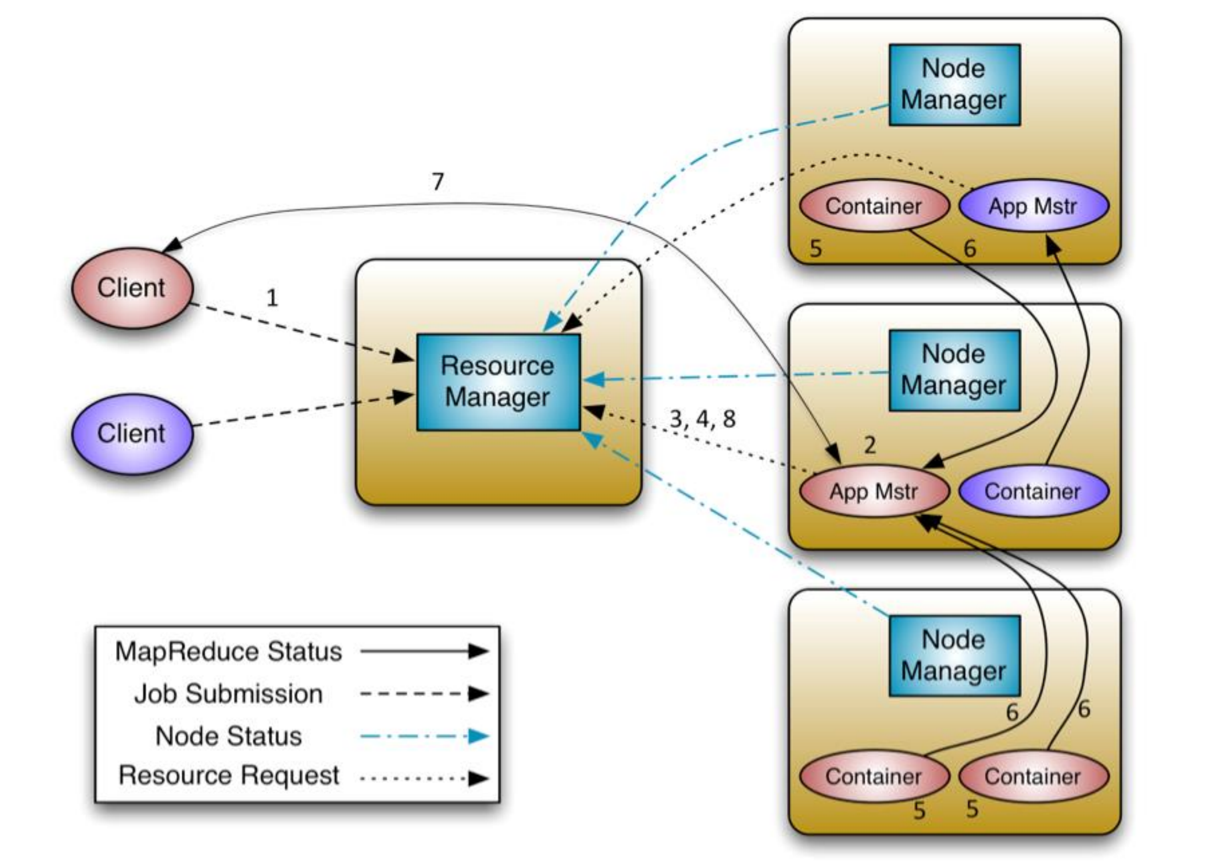
Applications Run Natively IN Hadoop

Pig	Hive	HBase	Accumulo	Storm	Solr	Spark	Cascading	Others
Script	SQL	NoSQL	NoSQL	Stream	Search	In-Memory	Java	ISV Engines

YARN: Data Operating System

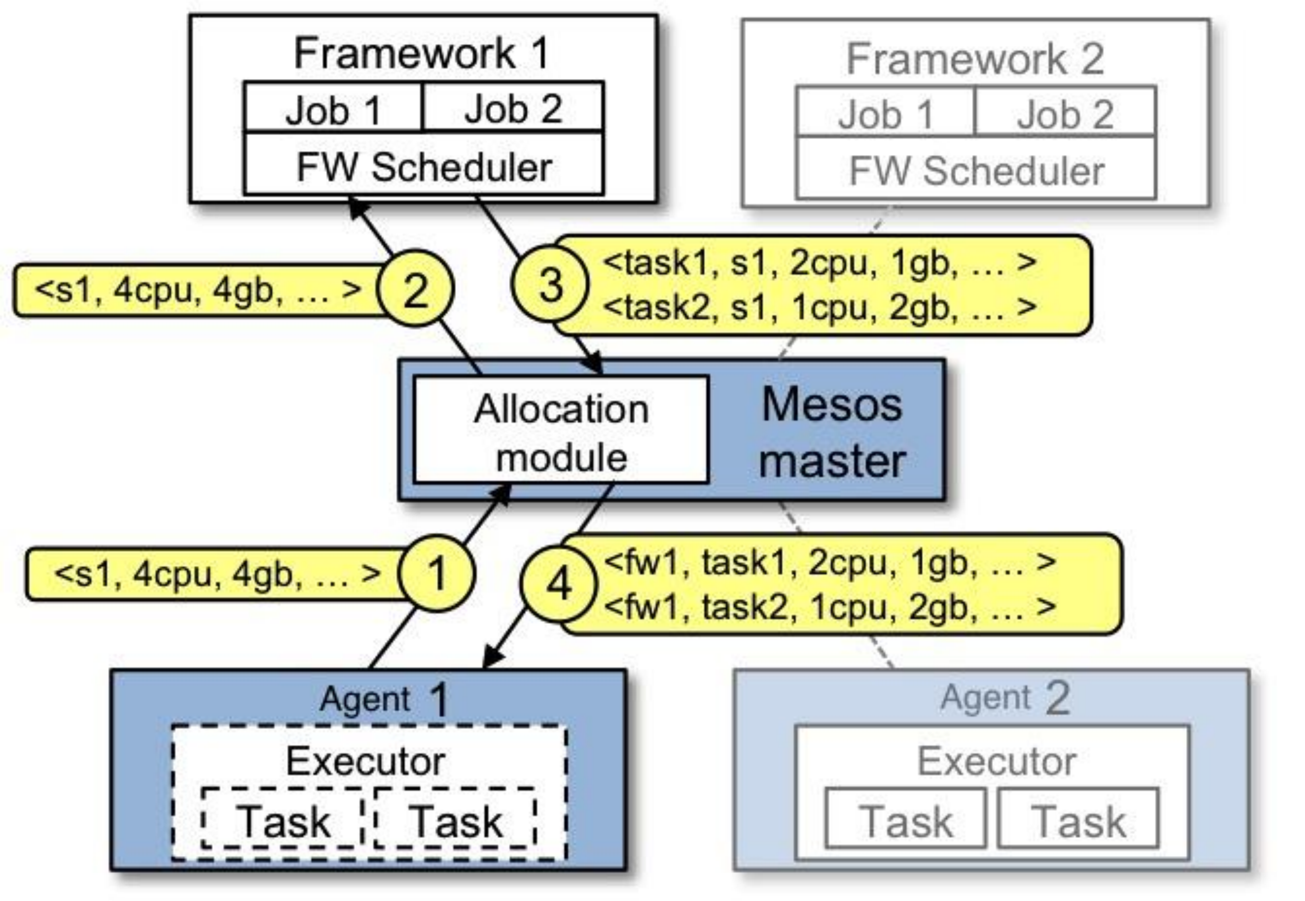
HDFS
(Hadoop Distributed File System)

Yarn



https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/WritingYarnApplications.html

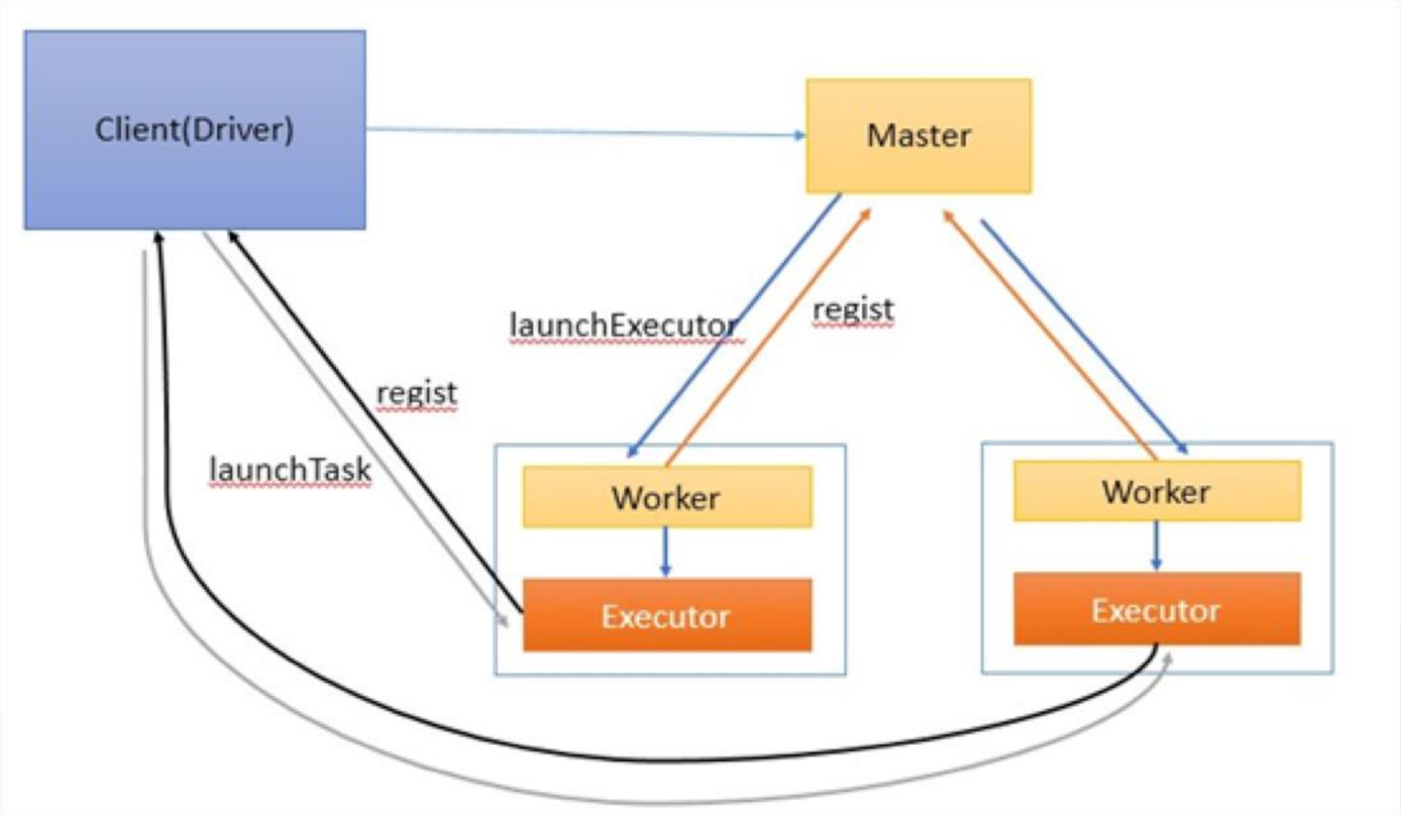
Mesos


Framework 1
Job 1
Job 2
FW Scheduler
Framework 2
Job 1
Job 2
FW Scheduler
<s1, 4cpu, 4gb, … >
2
3
<task1, s1, 2cpu, 1gb, … >
<task2, s1, 1cpu, 2gb, … >
Allocation module
Mesos master
<s1, 4cpu, 4gb, … >
1
4
<fw1, task1, 2cpu, 1gb, … >
<fw1, task2, 1cpu, 2gb, … >
Agent 1
Executor
Task
Task
Agent 2
Executor
Task
Task

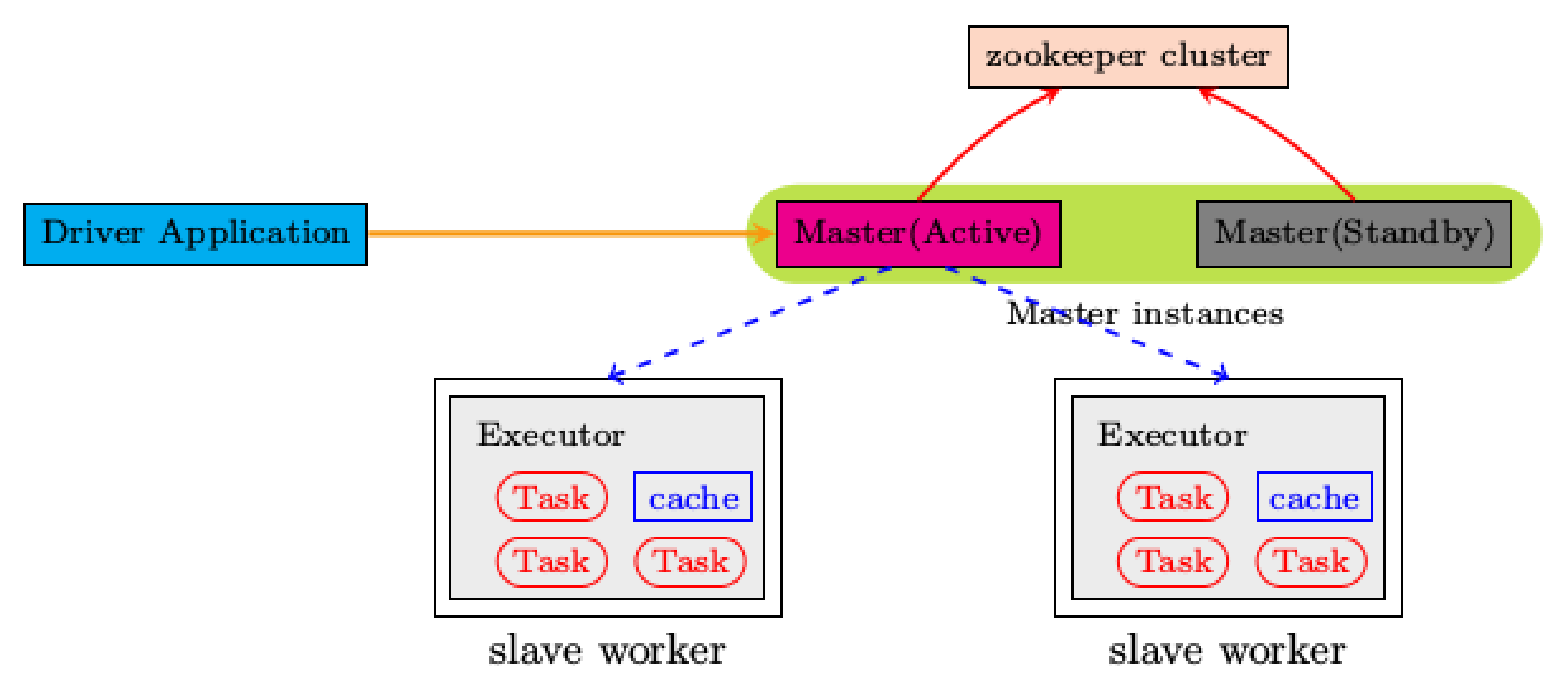
PART2

Spark Environment

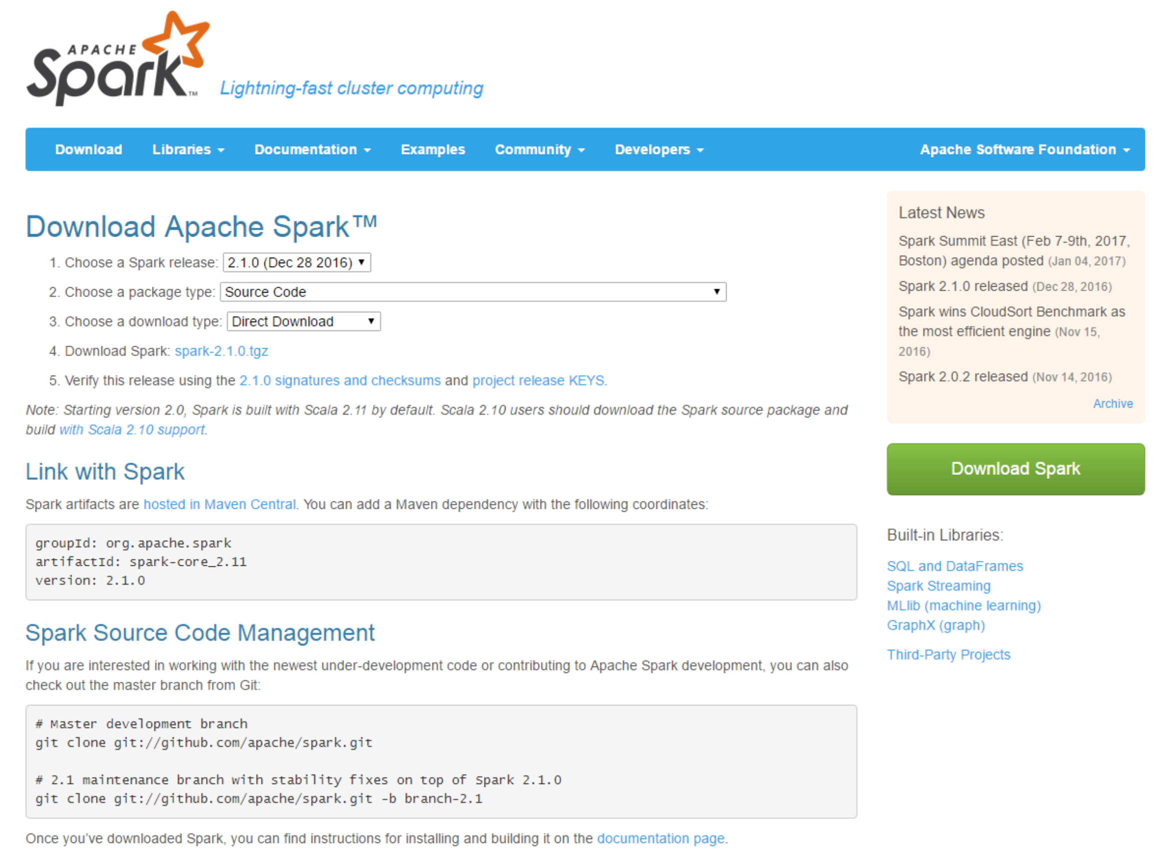
Spark Standalone



Spark Standalone HA



Spark Source Compile



Download Apache Spark™

1. Choose a Spark release: 2.1.0 (Dec 28 2016)
2. Choose a package type: Source Code
3. Choose a download type: Direct Download
4. Download Spark: spark-2.1.0.tgz
5. Verify this release using the 2.1.0 signatures and checksums and project release KEYS.

Note: Starting version 2.0, Spark is built with Scala 2.11 by default. Scala 2.10 users should download the Spark source package and build with Scala 2.10 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_2.11
version: 2.1.0
```

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:

```
# Master development branch
git clone git://github.com/apache/spark.git

# 2.1 maintenance branch with stability fixes on top of Spark 2.1.0
git clone git://github.com/apache/spark.git -b branch-2.1
```

Once you've downloaded Spark, you can find instructions for installing and building it on the documentation page.

Latest News

Spark Summit East (Feb 7-9th, 2017, Boston) agenda posted (Jan 04, 2017)

Spark 2.1.0 released (Dec 28, 2016)

Spark wins CloudSort Benchmark as the most efficient engine (Nov 15, 2016)

Spark 2.0.2 released (Nov 14, 2016)

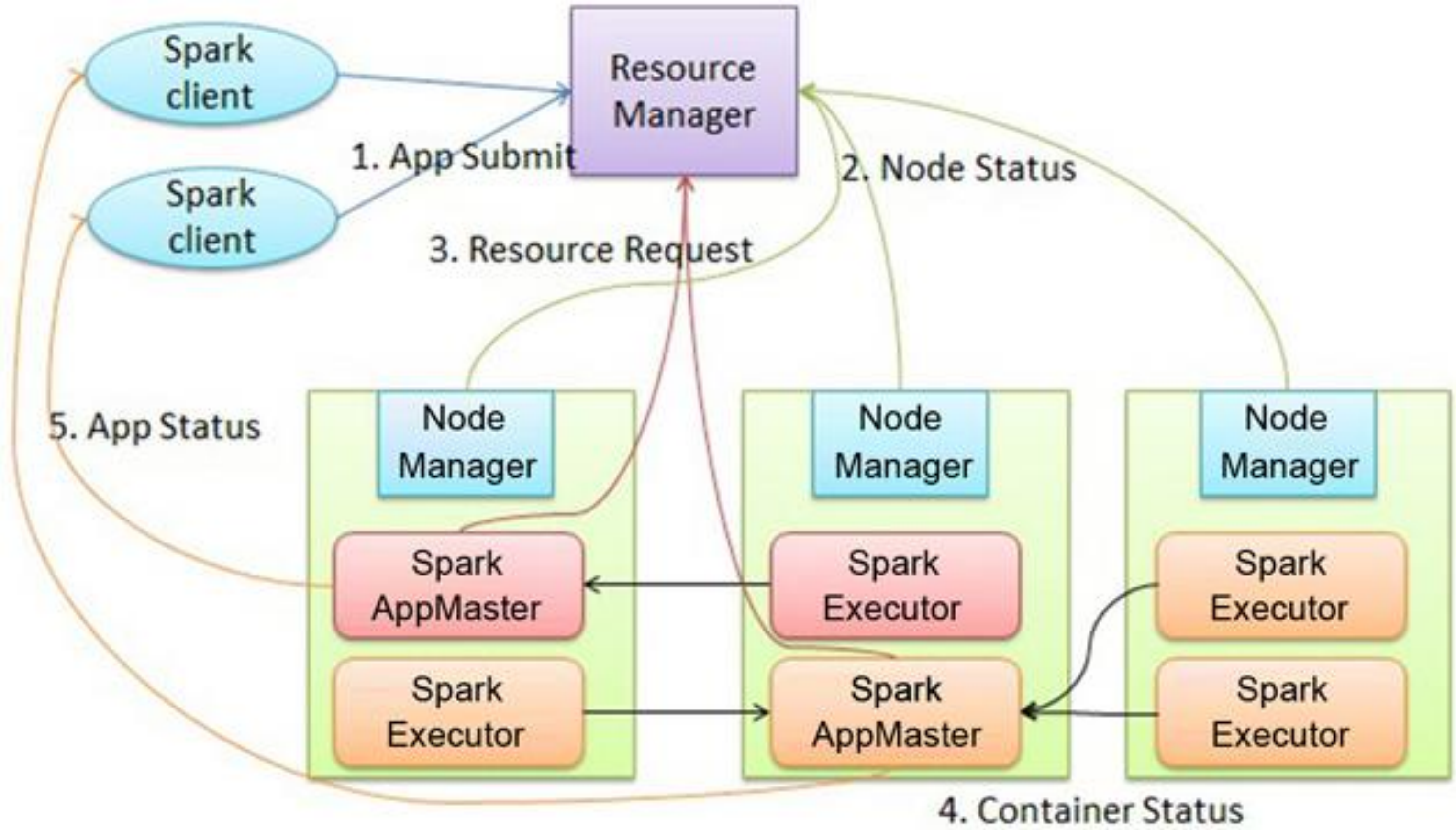
Archive

Built-in Libraries:

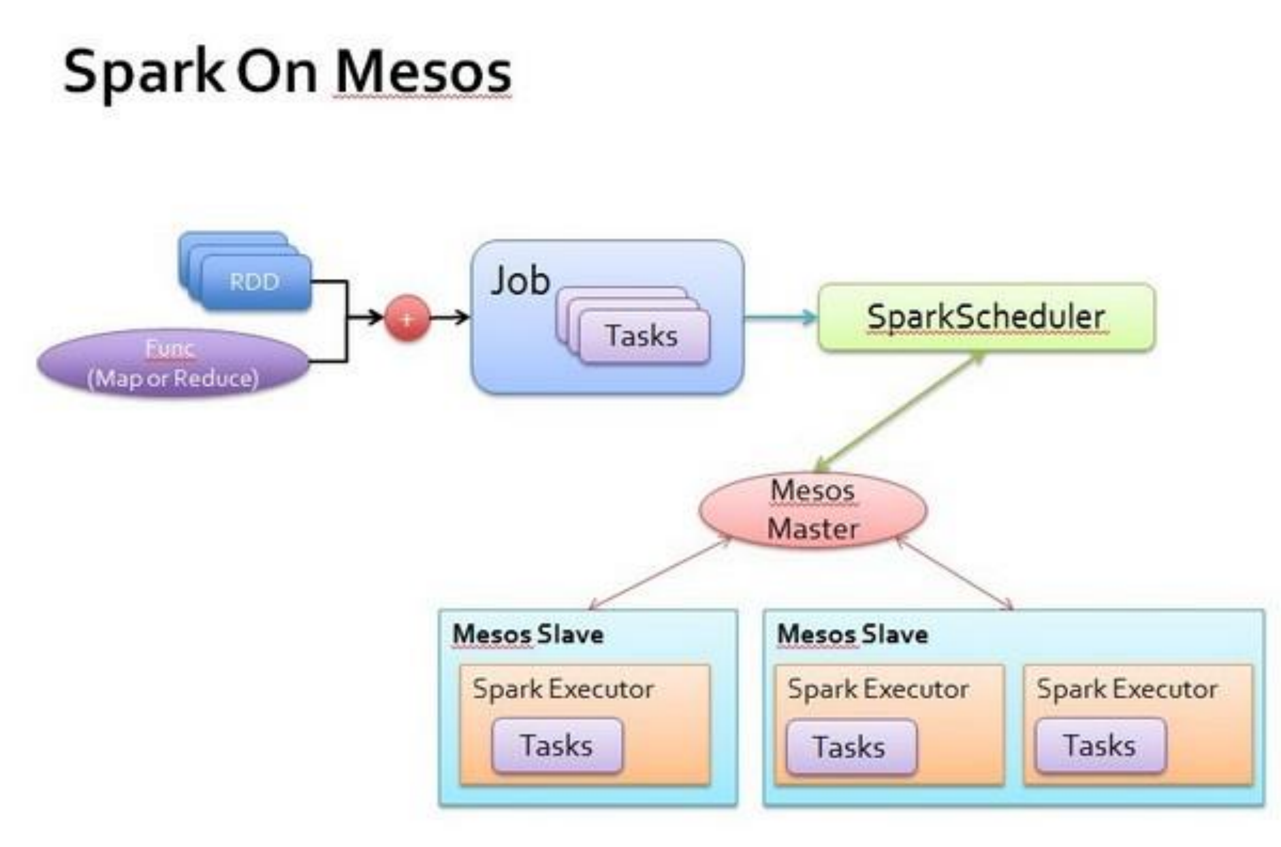
SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Projects

Spark On Yarn



Spark On Mesos

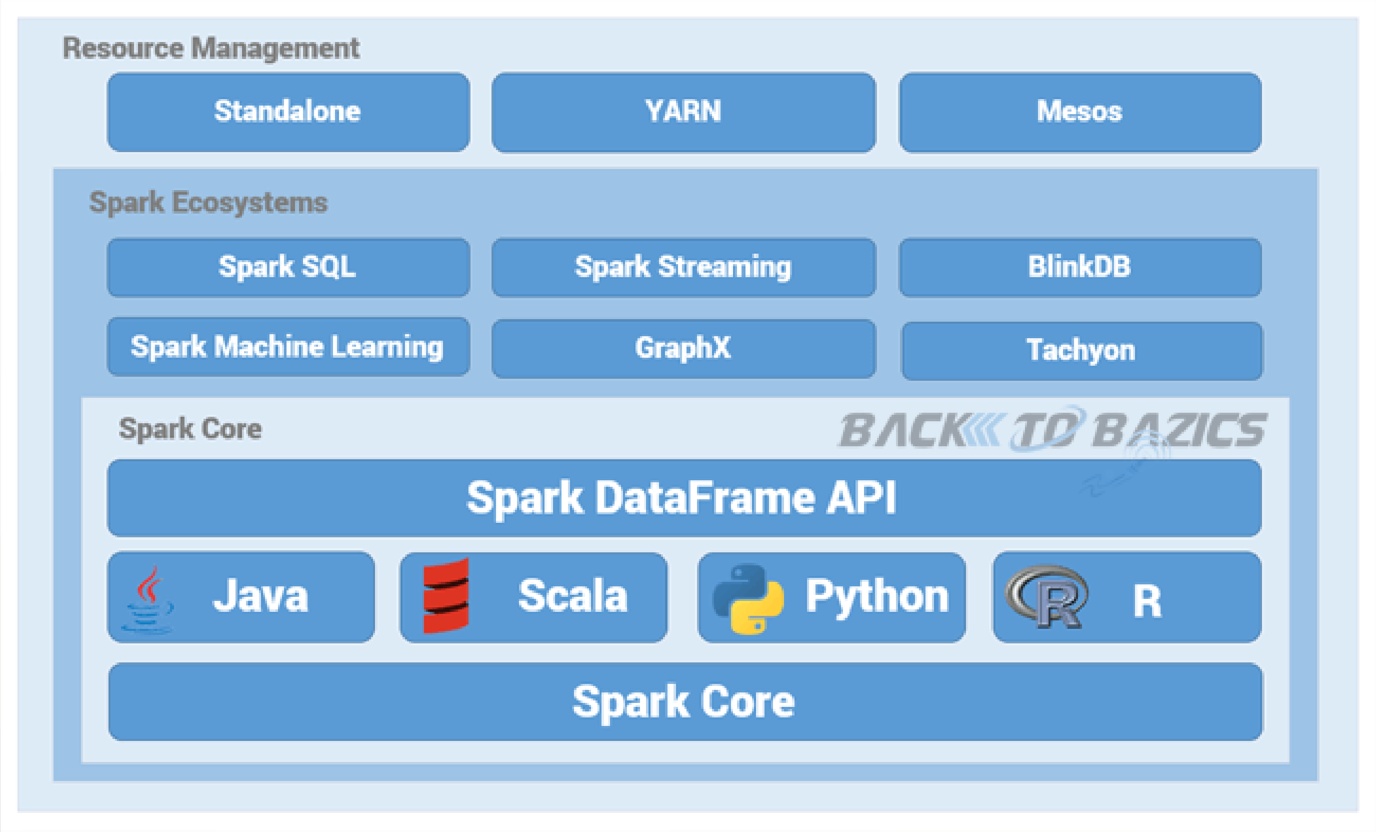


Spark On Mesos

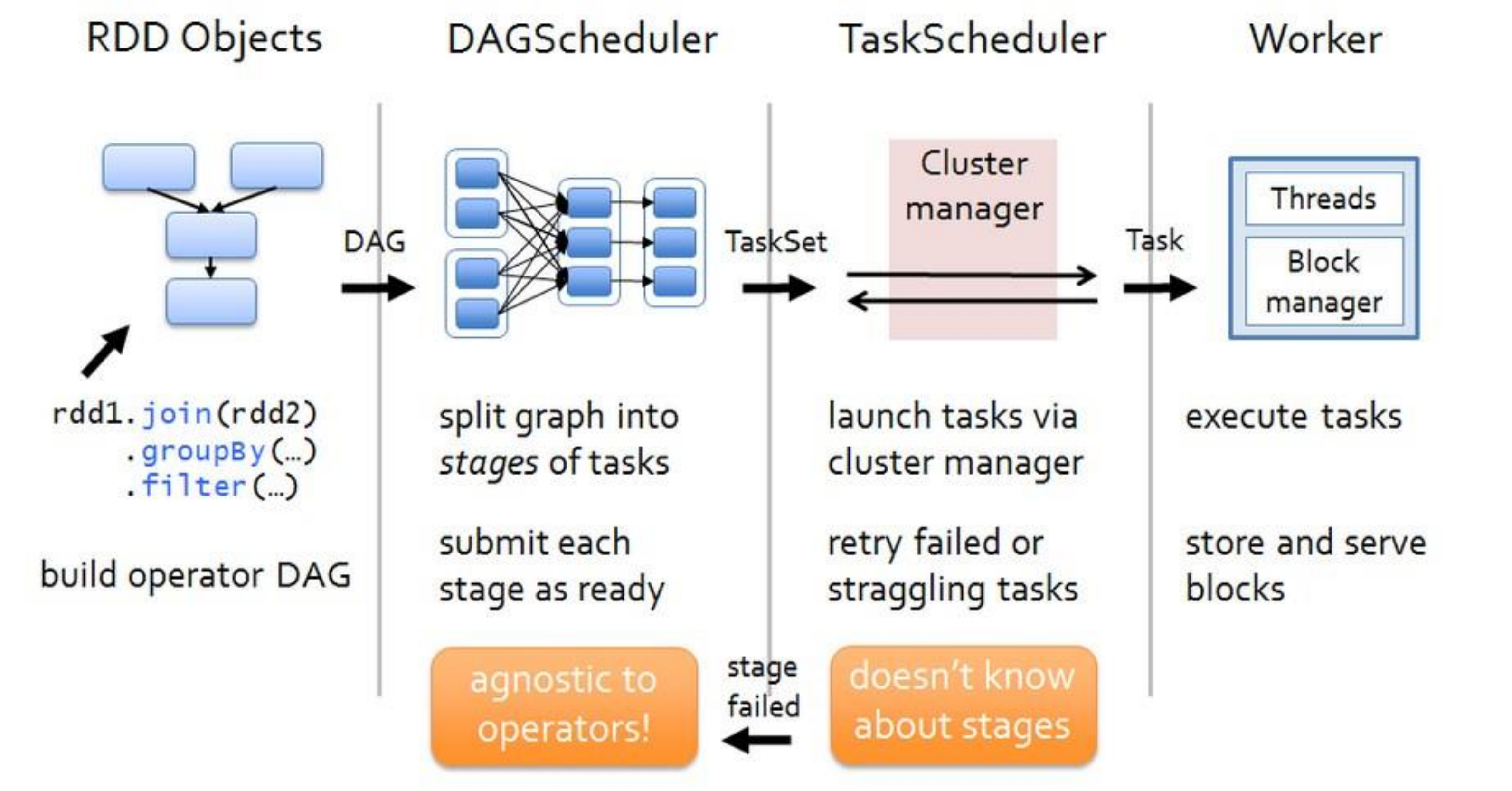
PART3

Spark Architecture

Ecosystems


Resource Management
Standalone
YARN
Mesos
Spark Ecosystems
Spark SQL
Spark Streaming
BlinkDB
Spark Machine Learning
GraphX
Tachyon
Spark Core
BACK TO BAZICS
Spark DataFrame API
Java
Scala
Python
R
Spark Core

Architecture



```
rdd1.join(rdd2)
    .groupBy(…)
    .filter(…)
```

build operator DAG

split graph into *stages* of tasks

submit each stage as ready

agnostic to operators!

launch tasks via cluster manager

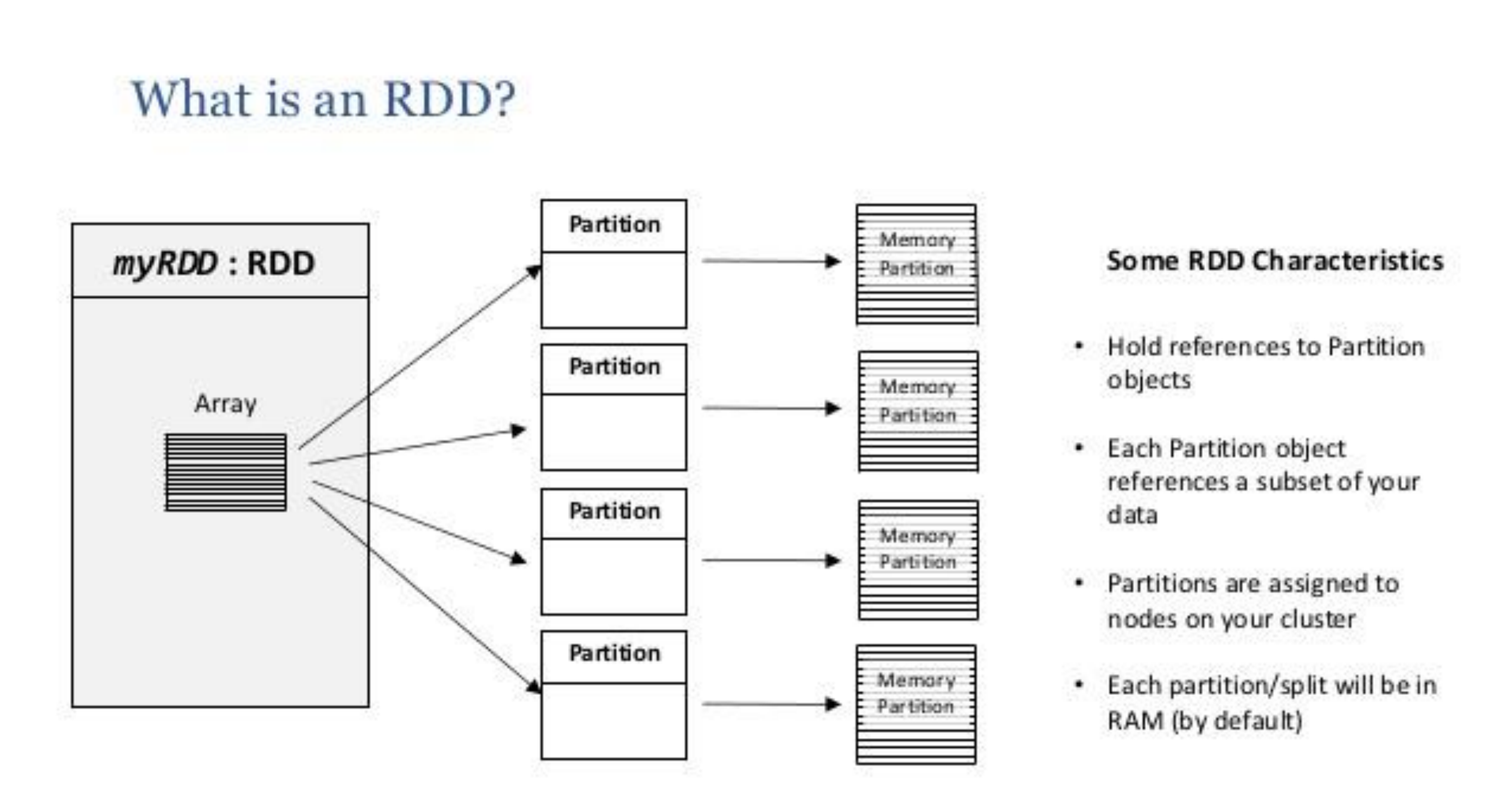
retry failed or straggling tasks

doesn't know about stages

execute tasks

store and serve blocks

RDD

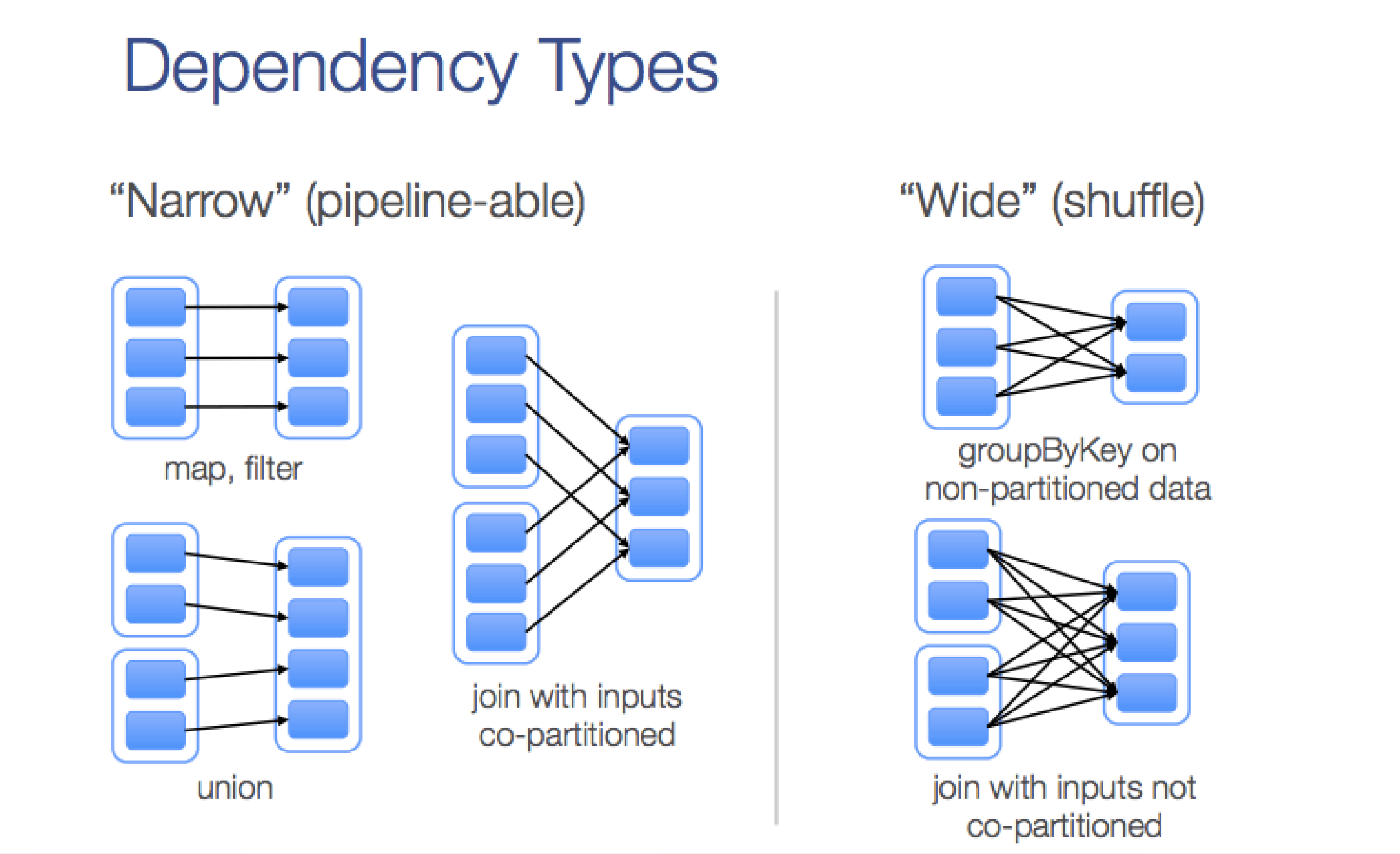


What is an RDD?

Some RDD Characteristics

- Hold references to Partition objects
- Each Partition object references a subset of your data
- Partitions are assigned to nodes on your cluster
- Each partition/split will be in RAM (by default)

Dependency Type



Dependency Types

“Narrow” (pipeline-able)

map, filter

union

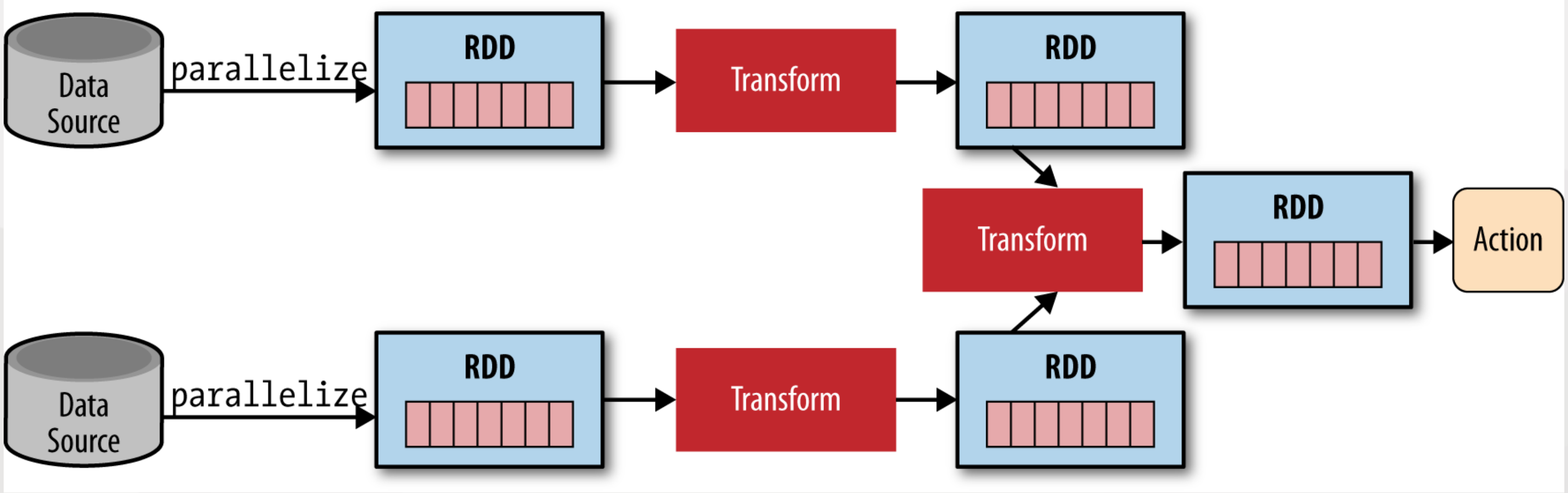
join with inputs co-partitioned

“Wide” (shuffle)

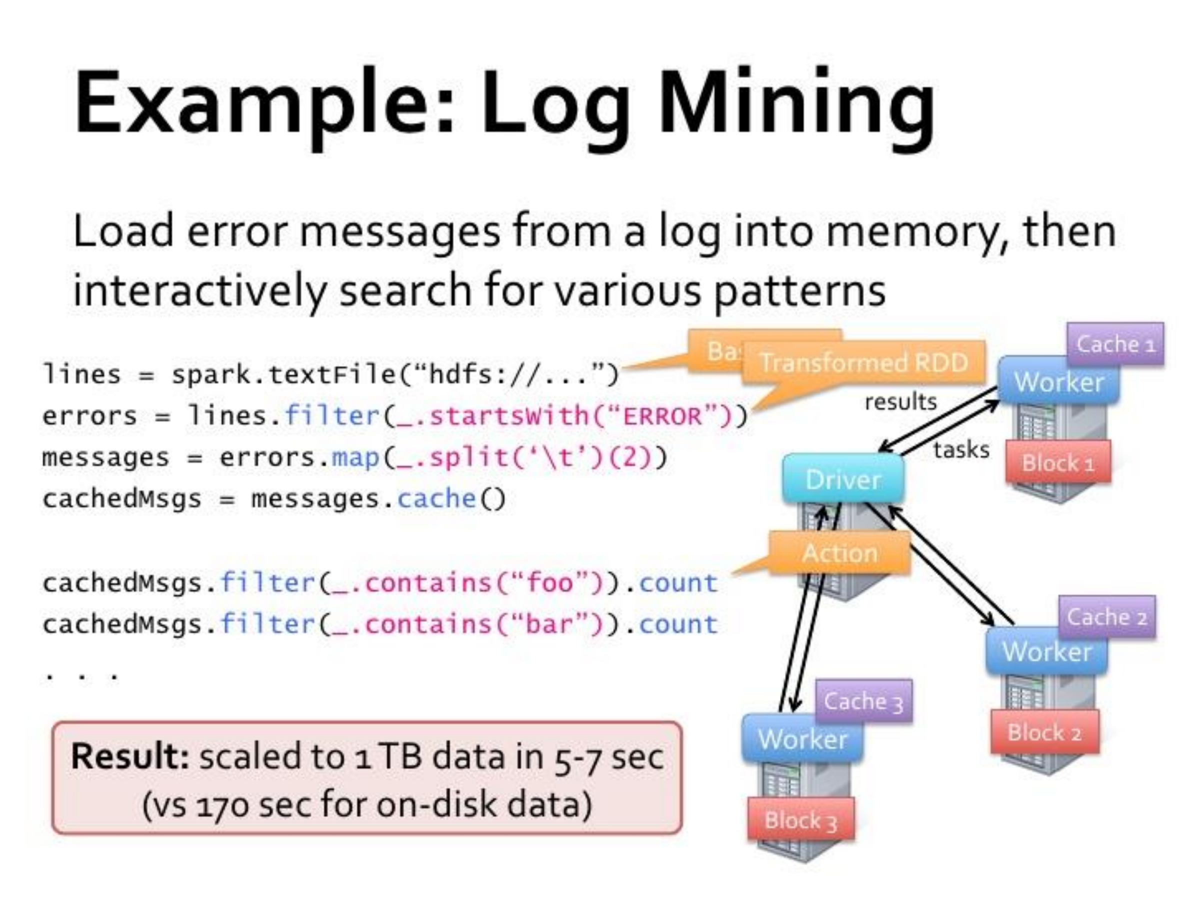
groupByKey on non-partitioned data

join with inputs not co-partitioned

Dependency Type



Example



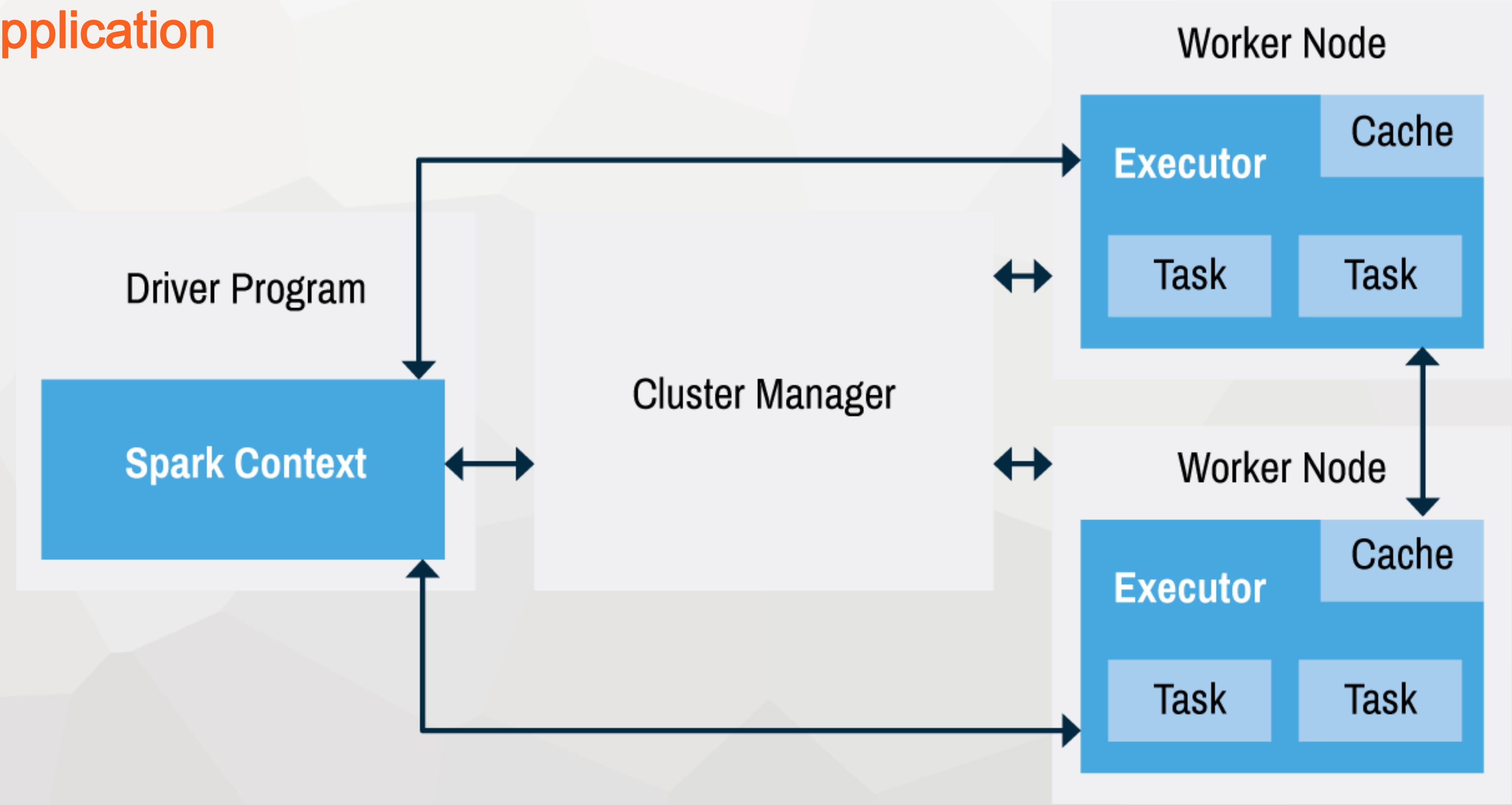
Example: Log Mining

Load error messages from a log into memory, then interactively search for various patterns

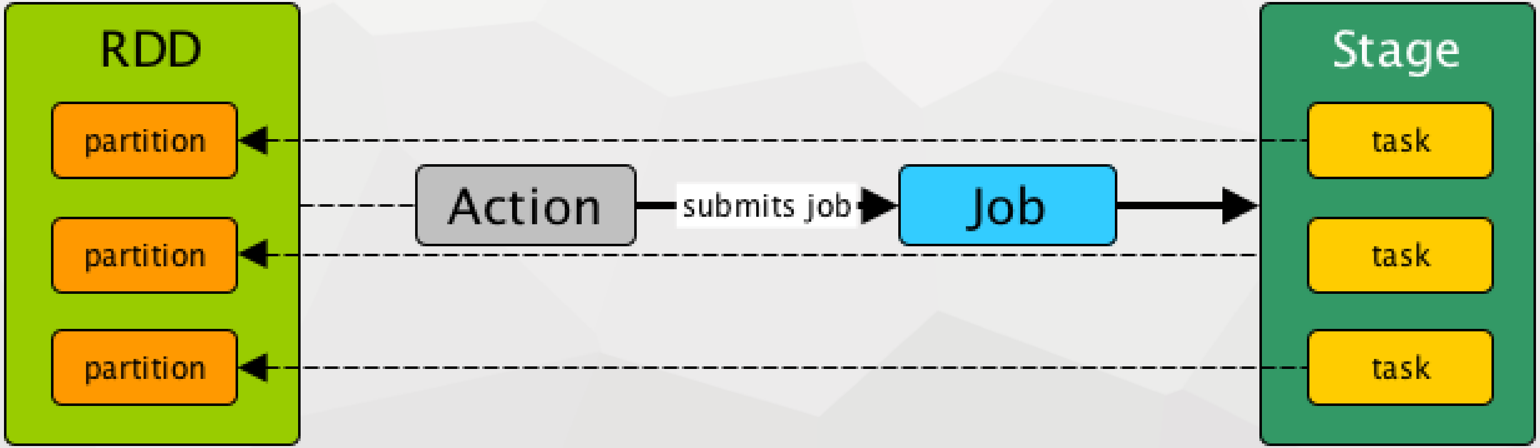
```
lines = spark.textFile("hdfs://...")
errors = lines.filter(_.startsWith("ERROR"))
messages = errors.map(_.split('\t')(2))
cachedMsgs = messages.cache()

cachedMsgs.filter(_.contains("foo")).count
cachedMsgs.filter(_.contains("bar")).count
. . .
```

Result: scaled to 1 TB data in 5-7 sec (vs 170 sec for on-disk data)

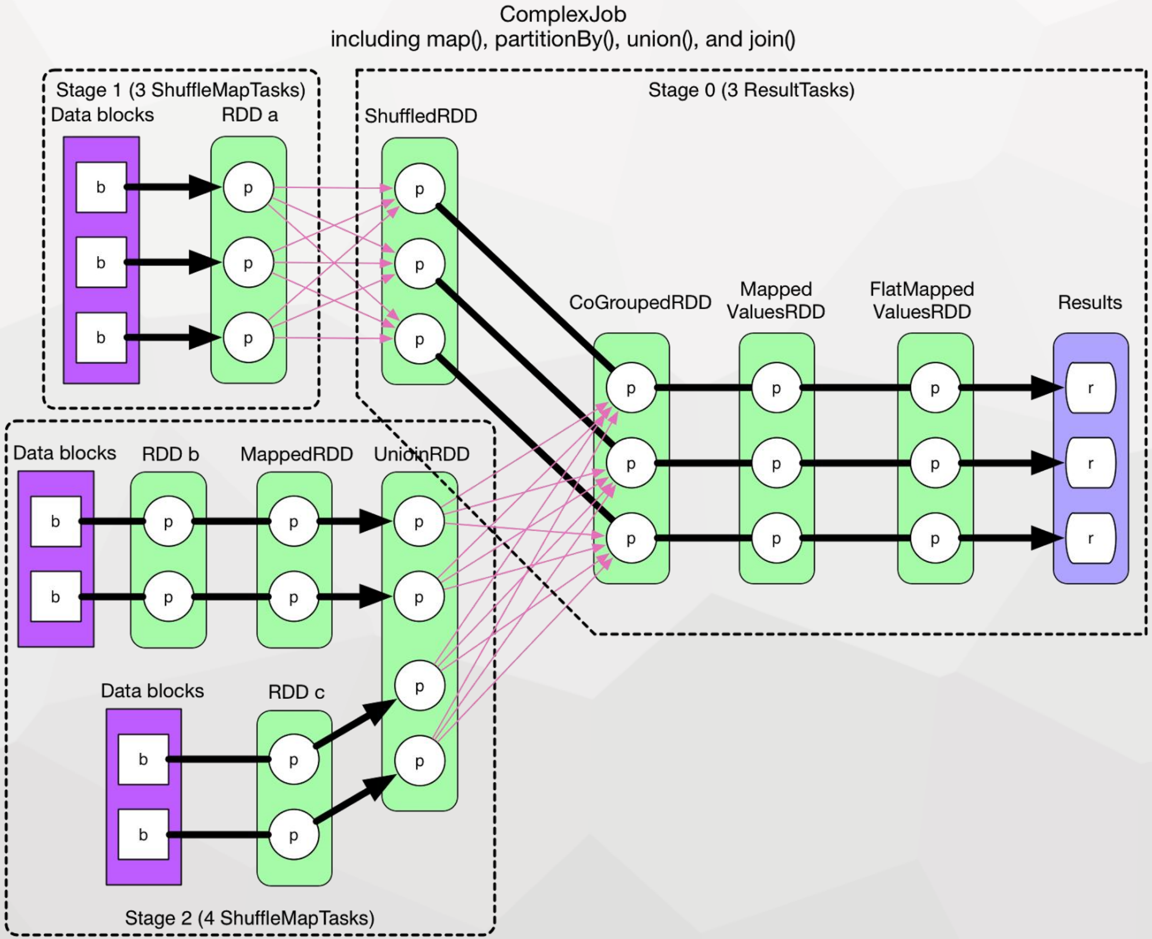
Driver
Executor
Application



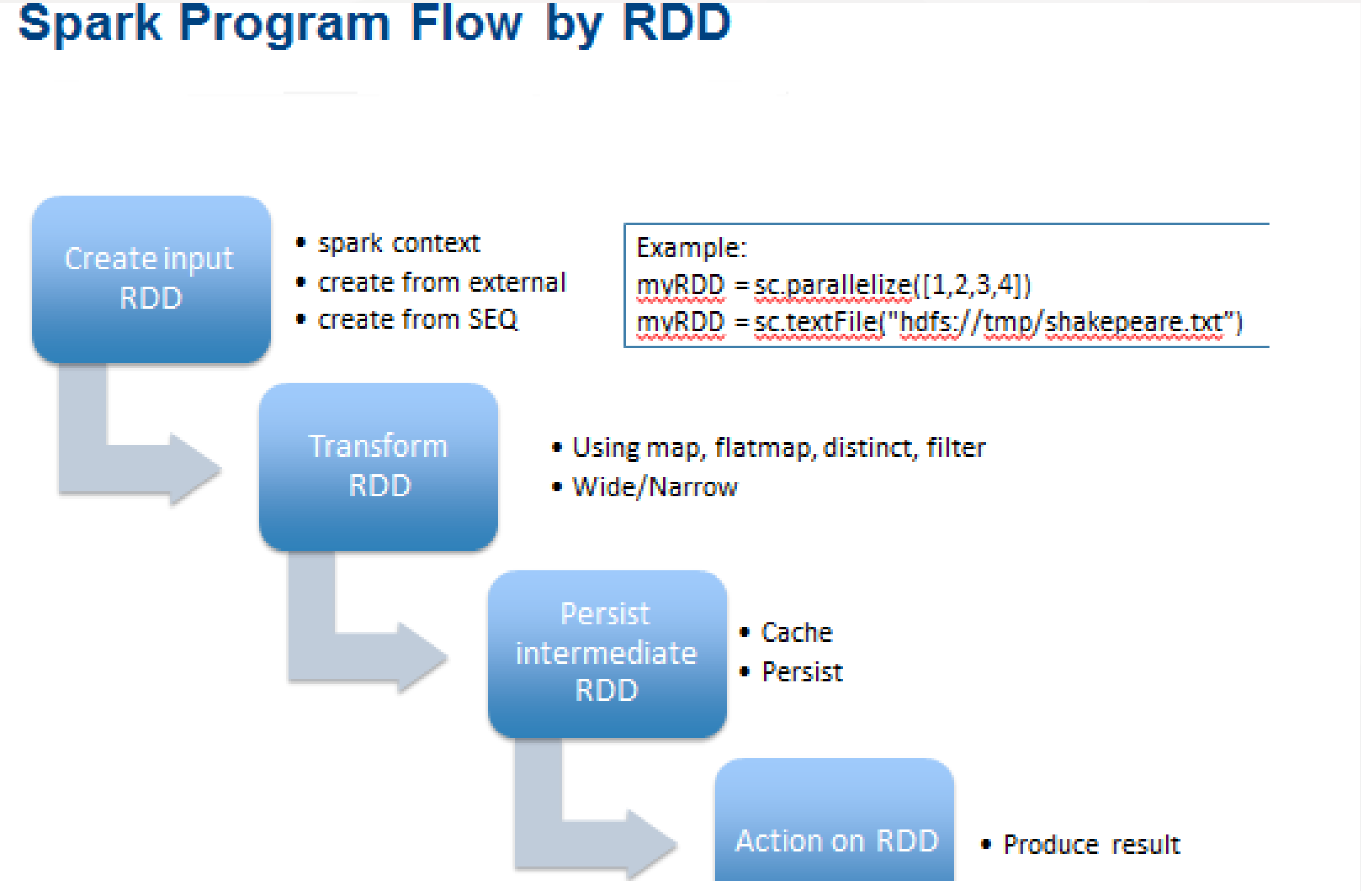
Job


RDD
partition
partition
partition
Action
submits job
Job
Stage
task
task
task

Stage


ComplexJob
including map(), partitionBy(), union(), and join()
Stage 1 (3 ShuffleMapTasks)
Data blocks
RDD a
b
p
Stage 0 (3 ResultTasks)
ShuffledRDD
CoGroupedRDD
Mapped ValuesRDD
FlatMapped ValuesRDD
Results
r
Data blocks
RDD b
MappedRDD
UnionRDD
Data blocks
RDD c
Stage 2 (4 ShuffleMapTasks)

Persist & Cache



Spark Program Flow by RDD

Create input RDD

- spark context
- create from external
- create from SEQ

Example:

```
myRDD = sc.parallelize([1,2,3,4])
myRDD = sc.textFile("hdfs://tmp/shakepeare.txt")
```

Transform RDD

- Using map, flatmap, distinct, filter
- Wide/Narrow

Persist intermediate RDD

- Cache
- Persist

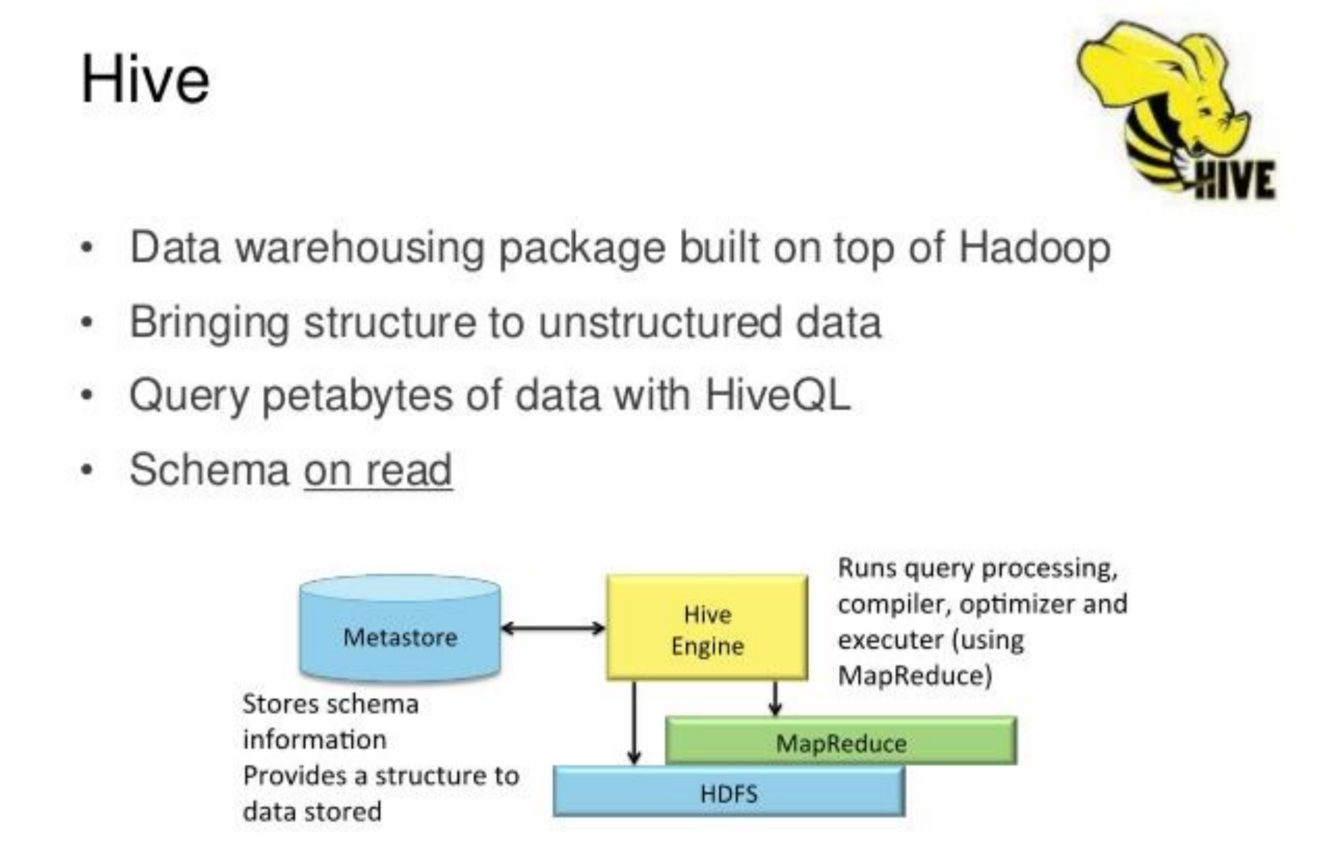
Action on RDD

- Produce result

PART4

Spark SQL

Hive



Hive

- Data warehousing package built on top of Hadoop
- Bringing structure to unstructured data
- Query petabytes of data with HiveQL
- Schema on read

Metastore

Hive Engine

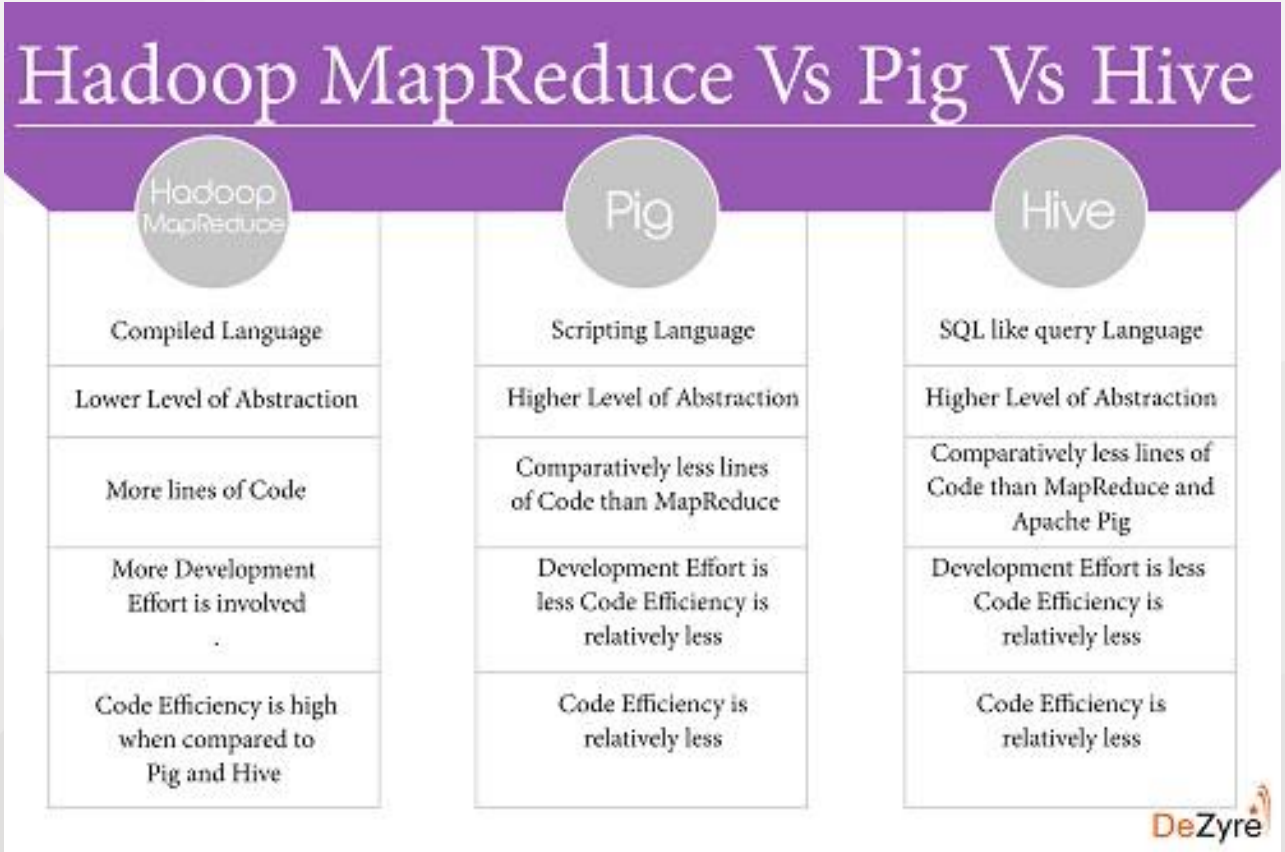
Runs query processing, compiler, optimizer and executer (using MapReduce)

MapReduce

HDFS

Stores schema information
Provides a structure to data stored

Hive

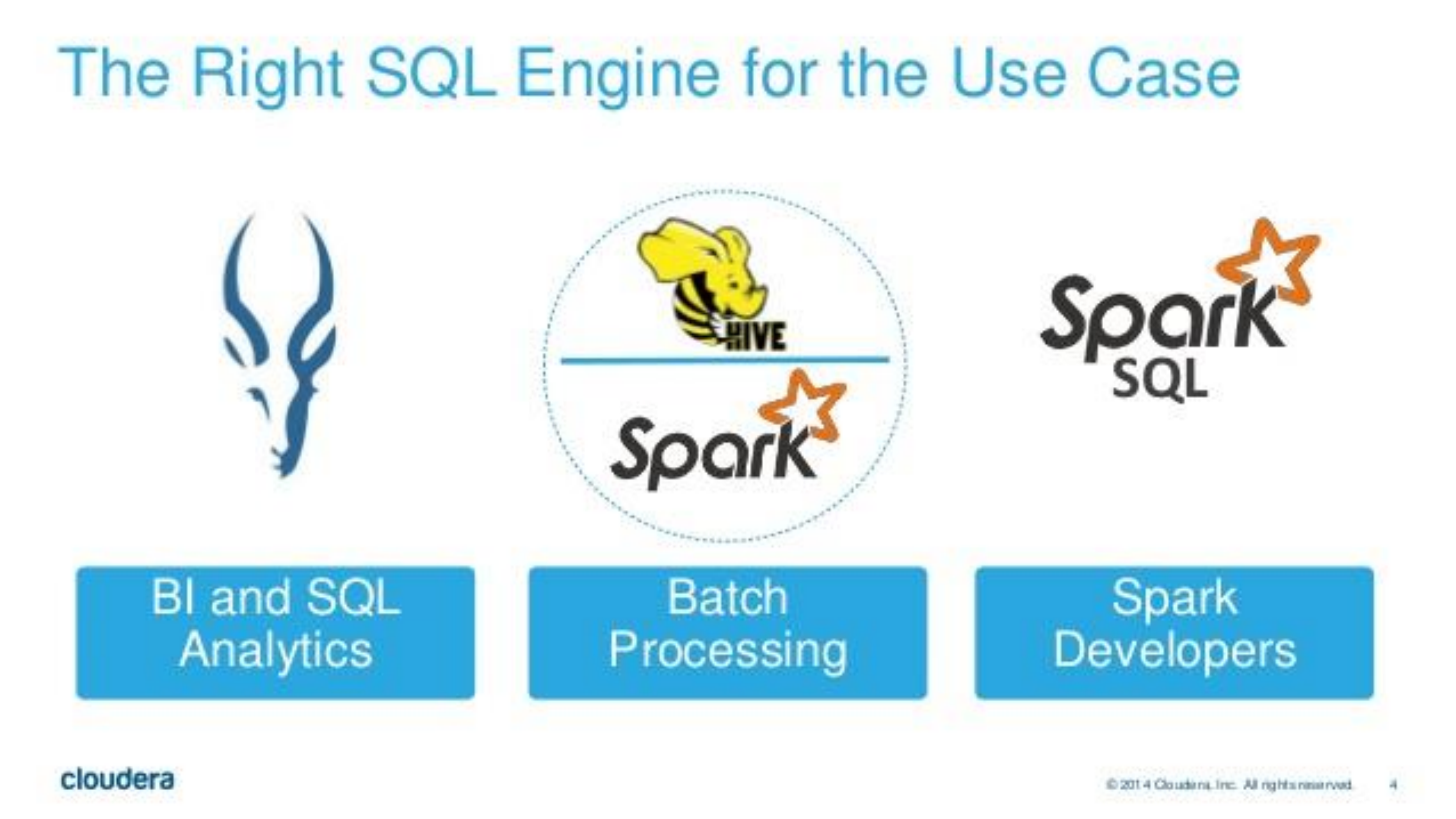


Hadoop MapReduce Vs Pig Vs Hive

Hadoop MapReduce	Pig	Hive
Compiled Language	Scripting Language	SQL like query Language
Lower Level of Abstraction	Higher Level of Abstraction	Higher Level of Abstraction
More lines of Code	Comparatively less lines of Code than MapReduce	Comparatively less lines of Code than MapReduce and Apache Pig
More Development Effort is involved .	Development Effort is less Code Efficiency is relatively less	Development Effort is less Code Efficiency is relatively less
Code Efficiency is high when compared to Pig and Hive	Code Efficiency is relatively less	Code Efficiency is relatively less

DeZyre

Spark SQL



The Right SQL Engine for the Use Case

HIVE
Spark
Spark
SQL

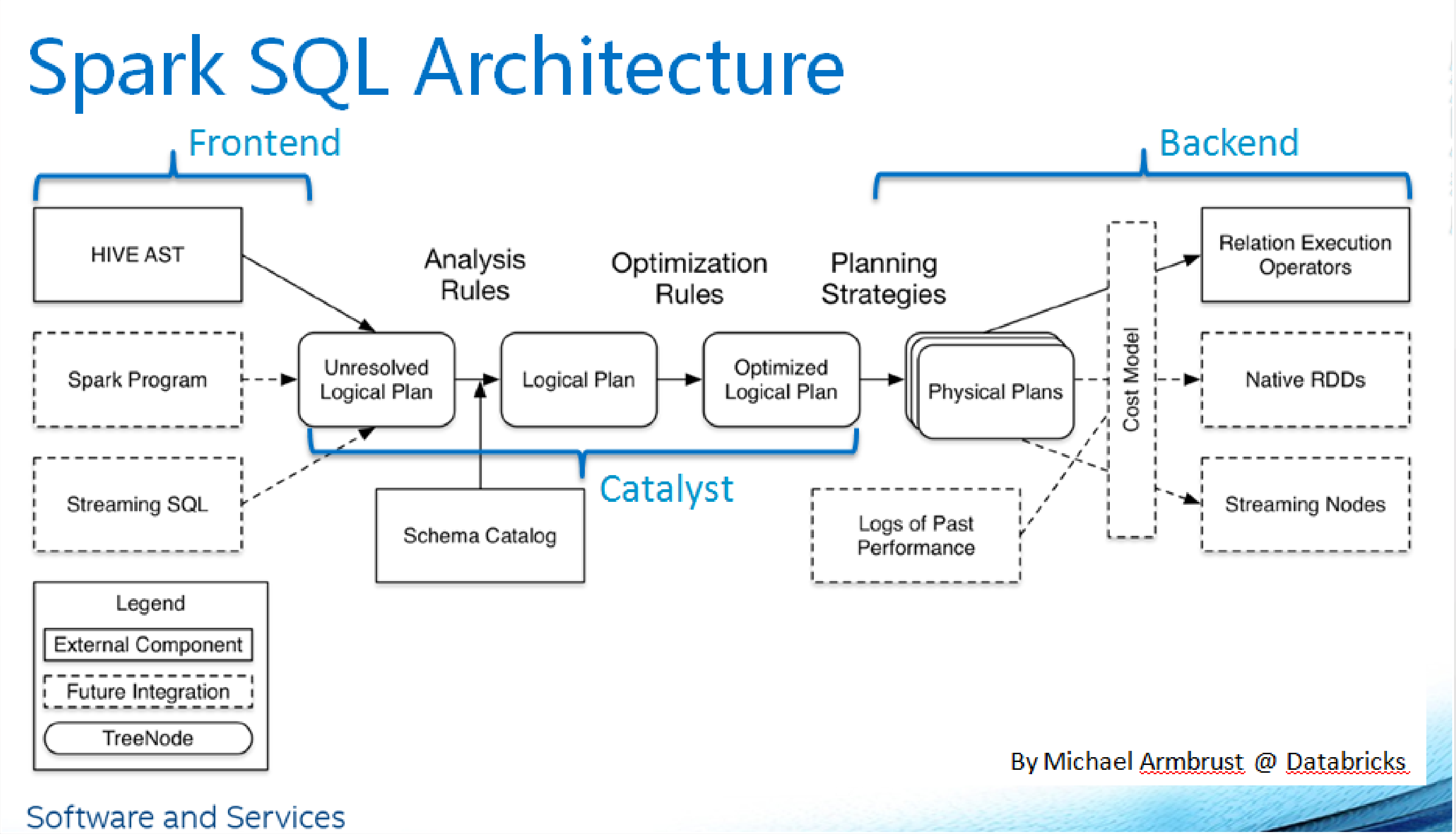
BI and SQL Analytics

Batch Processing

Spark Developers

cloudera

Spark SQL



Spark SQL Architecture

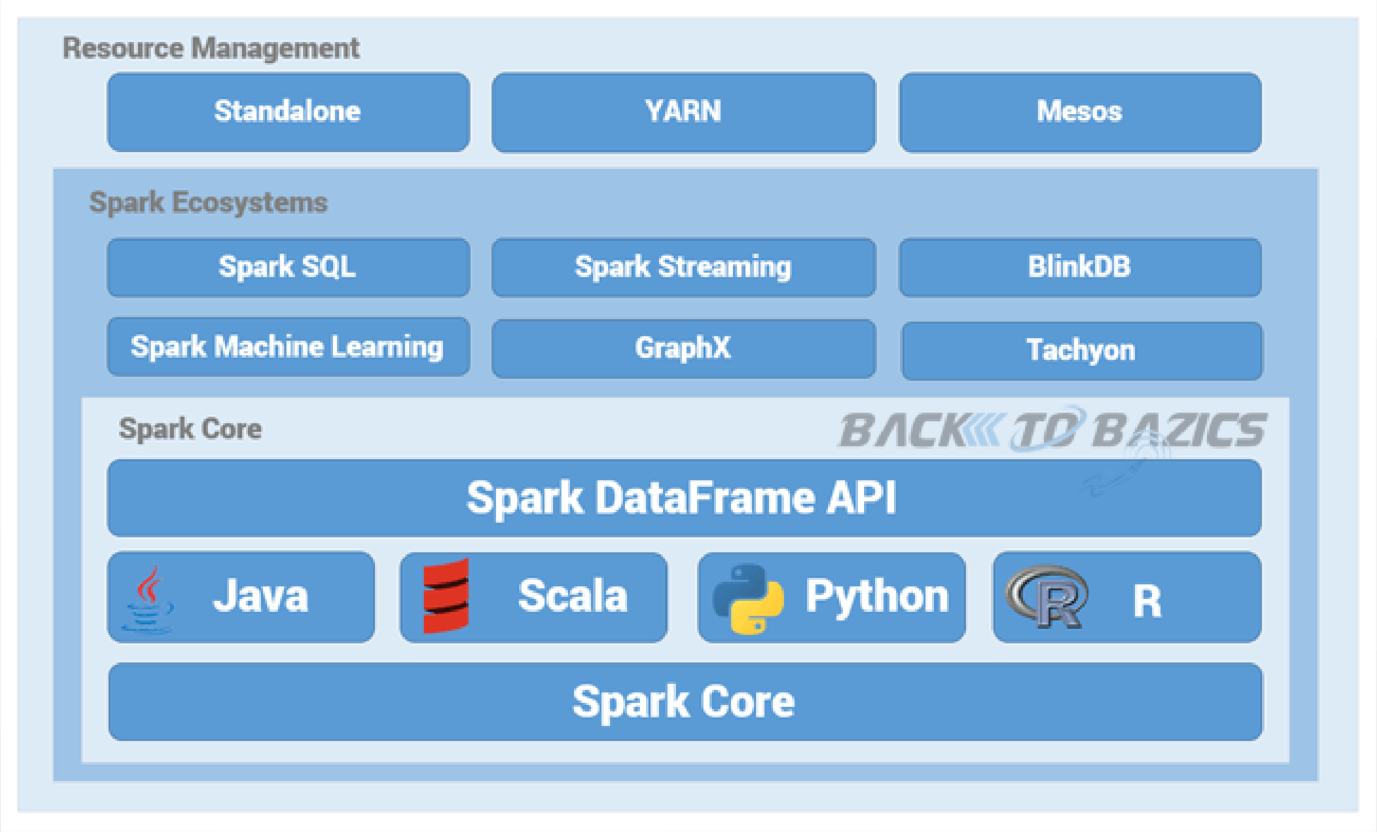
By Michael Armbrust @ Databricks

Software and Services

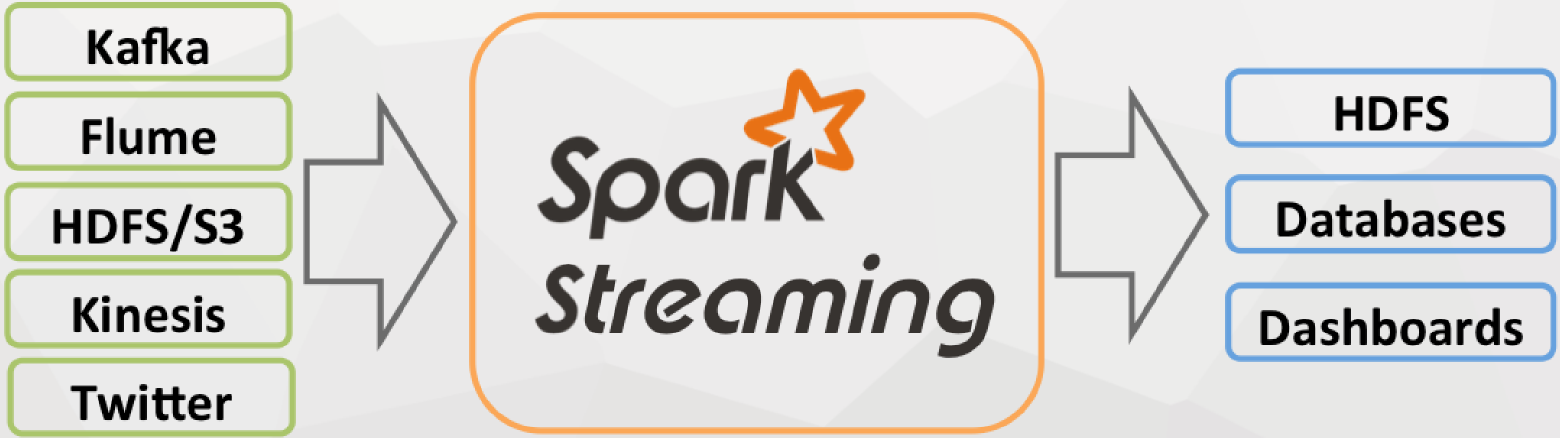
PART5

Spark Streaming

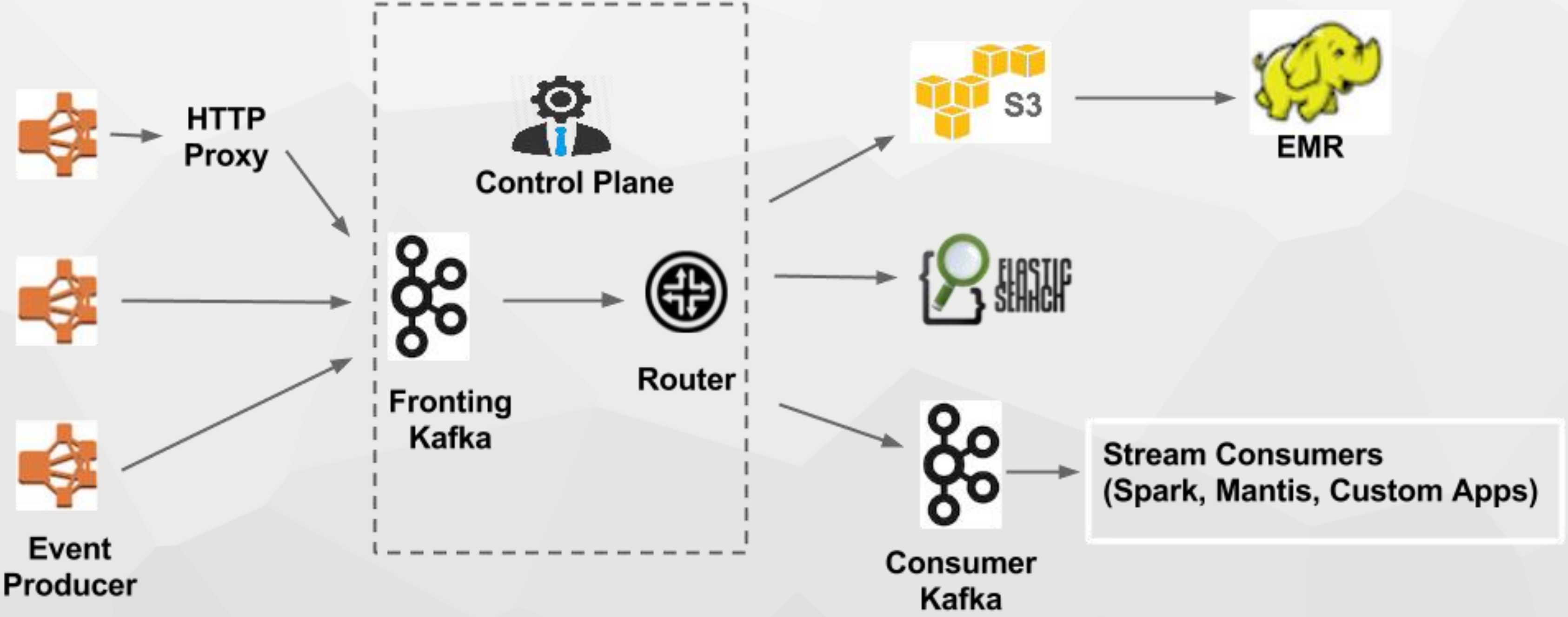
Ecosystems


Resource Management
Standalone
YARN
Mesos
Spark Ecosystems
Spark SQL
Spark Streaming
BlinkDB
Spark Machine Learning
GraphX
Tachyon
Spark Core
BACK TO BAZICS
Spark DataFrame API
Java
Scala
Python
R
Spark Core

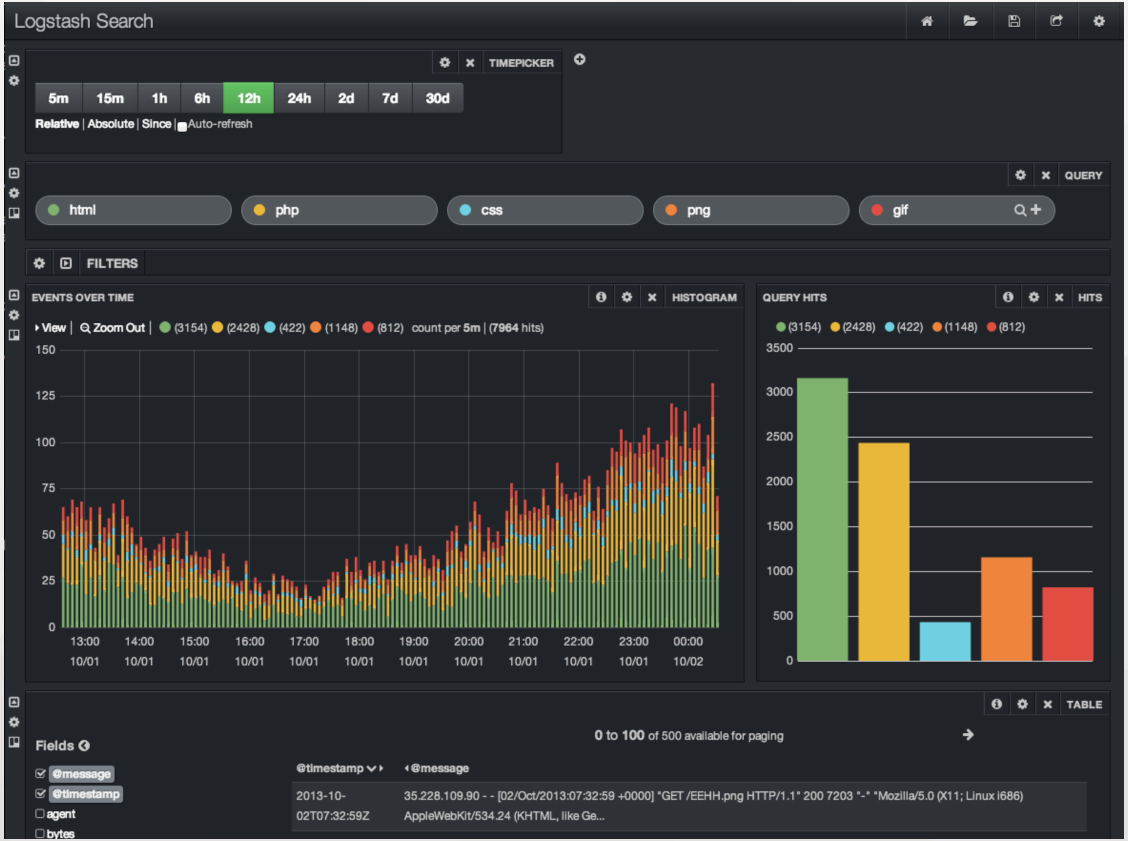
Spark Streaming



Kafka


HTTP Proxy
Control Plane
Fronting Kafka
Router
Event Producer
S3
EMR
ELASTIC SEARCH
Consumer Kafka
Stream Consumers (Spark, Mantis, Custom Apps)

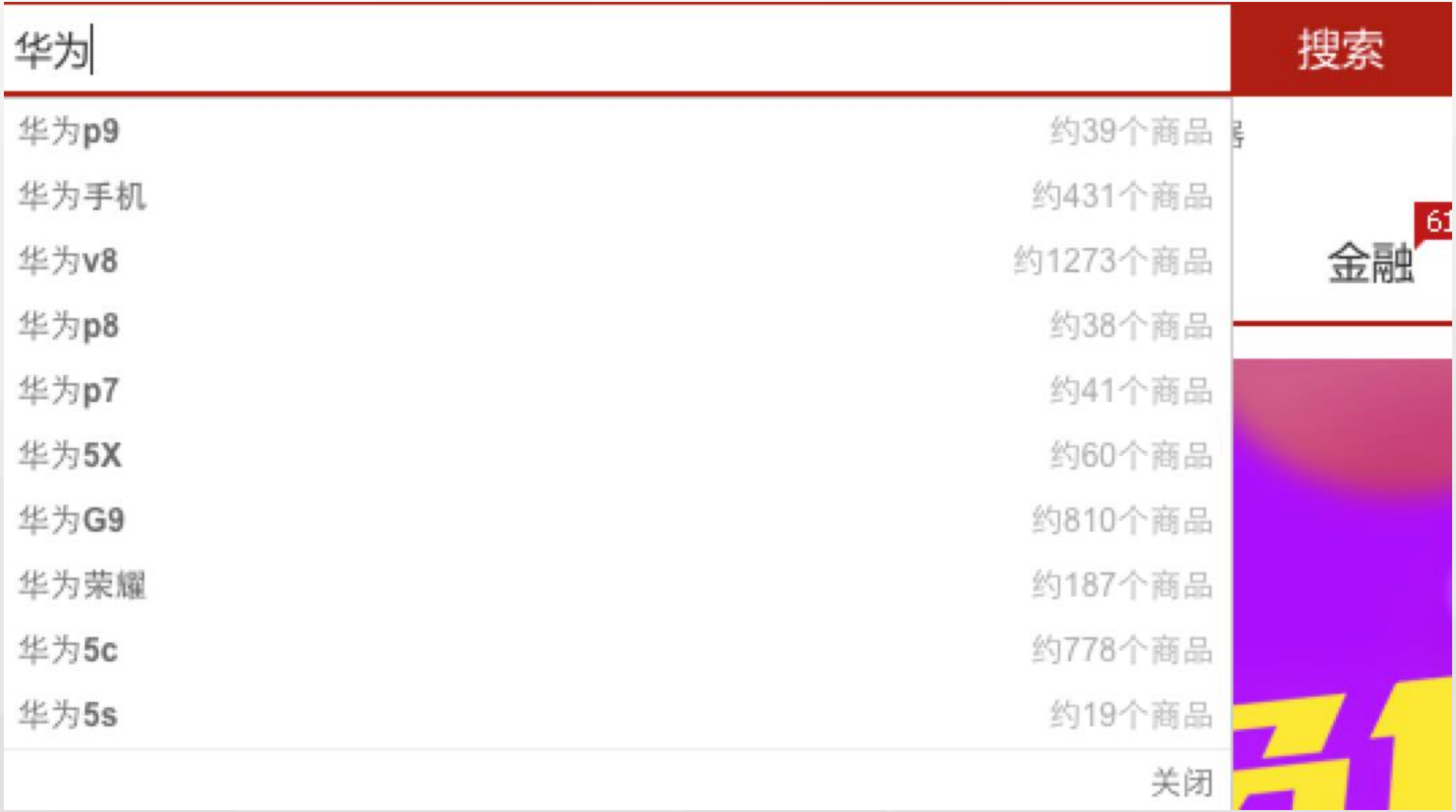
ELK (ElasticSearch LogStash Kibana)



0 to 100 of 500 available for paging

@timestamp	@message
2013-10-02T07:32:59Z	35.228.109.90 - - [02/Oct/2013:07:32:59 +0000] "GET /EEHH.png HTTP/1.1" 200 7203 "-" "Mozilla/5.0 (X11; Linux i686) AppleWebKit/534.24 (KHTML, like Ge...

Search Engine

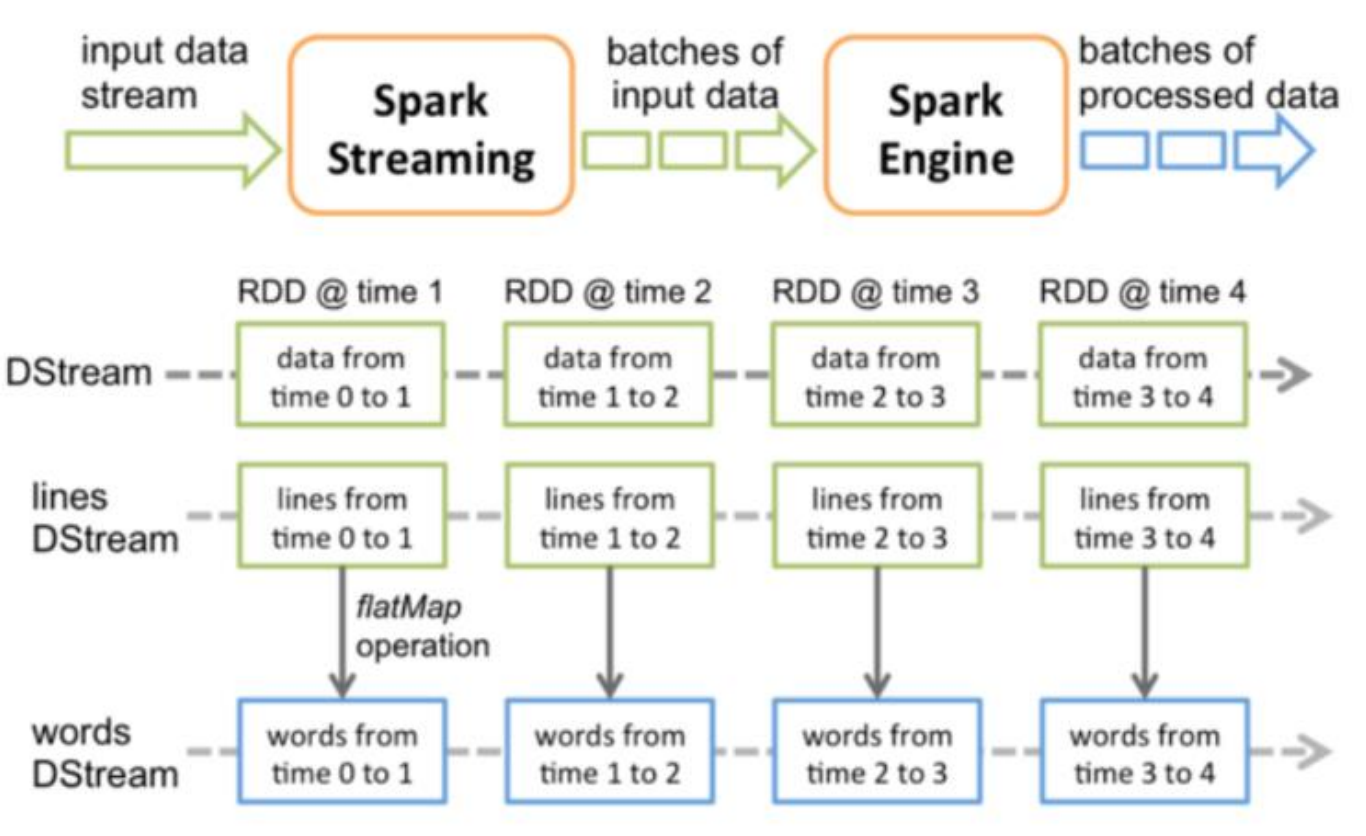


华为

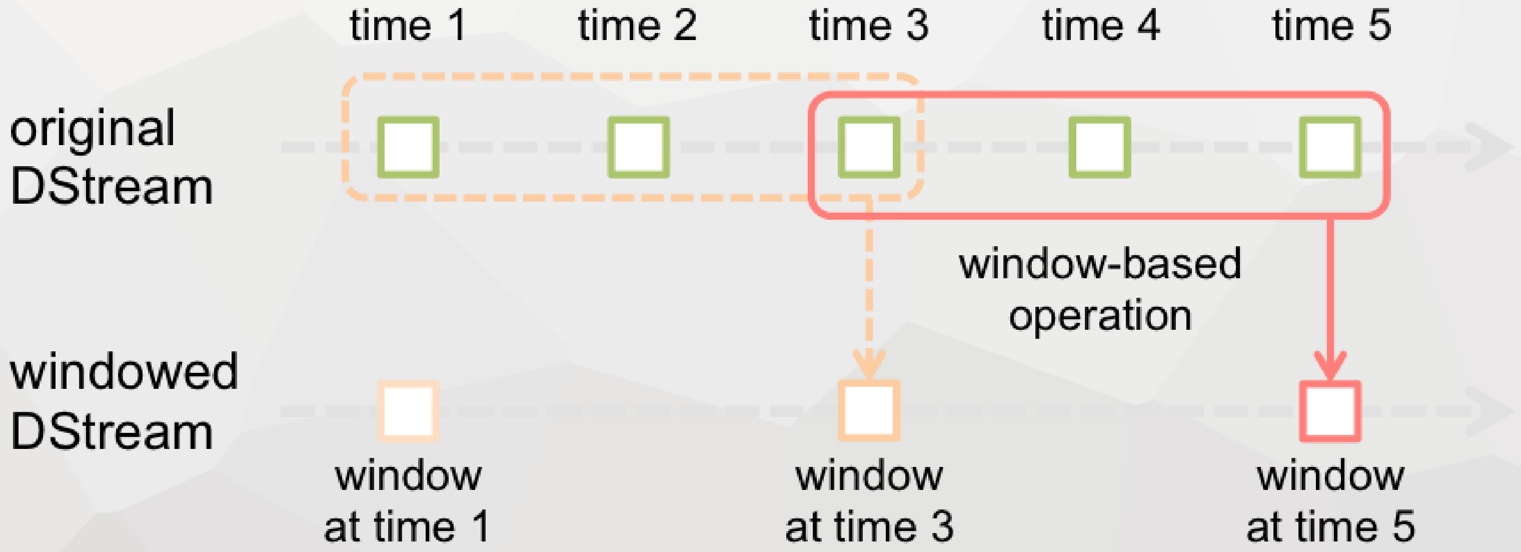
华为p9	约39个商品
华为手机	约431个商品
华为v8	约1273个商品
华为p8	约38个商品
华为p7	约41个商品
华为5X	约60个商品
华为G9	约810个商品
华为荣耀	约187个商品
华为5c	约778个商品
华为5s	约19个商品

关闭

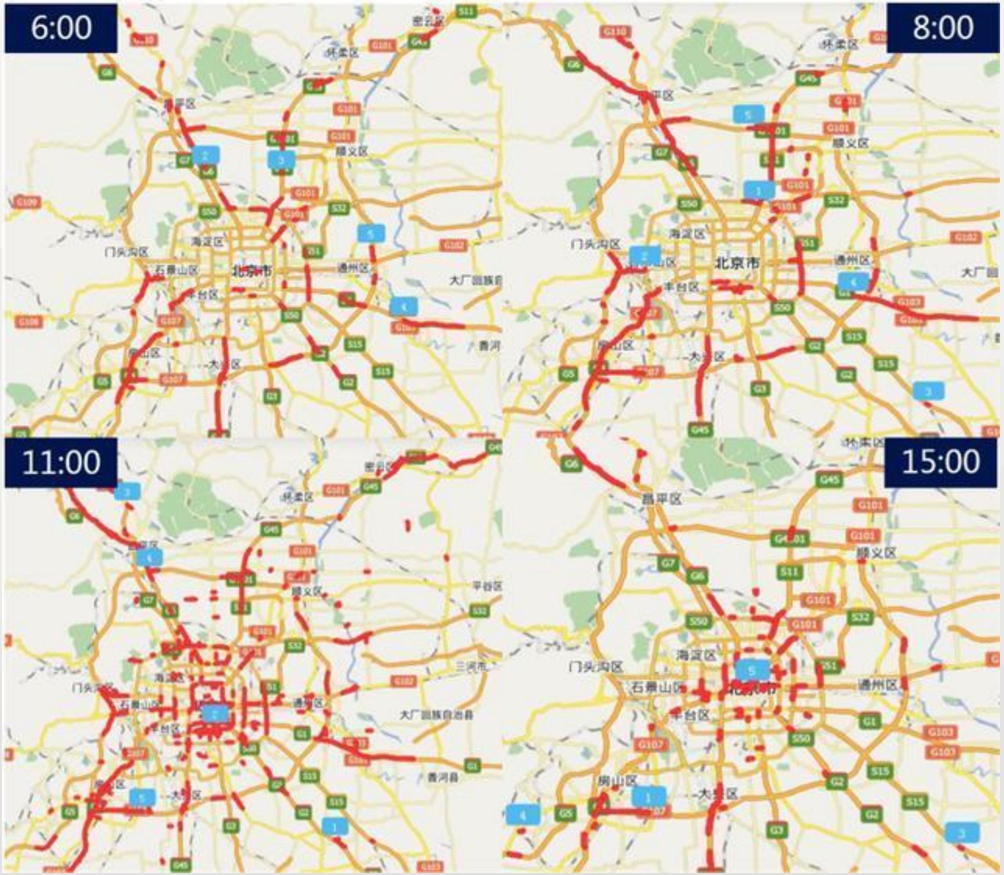
Spark Streaming



Windows based operations



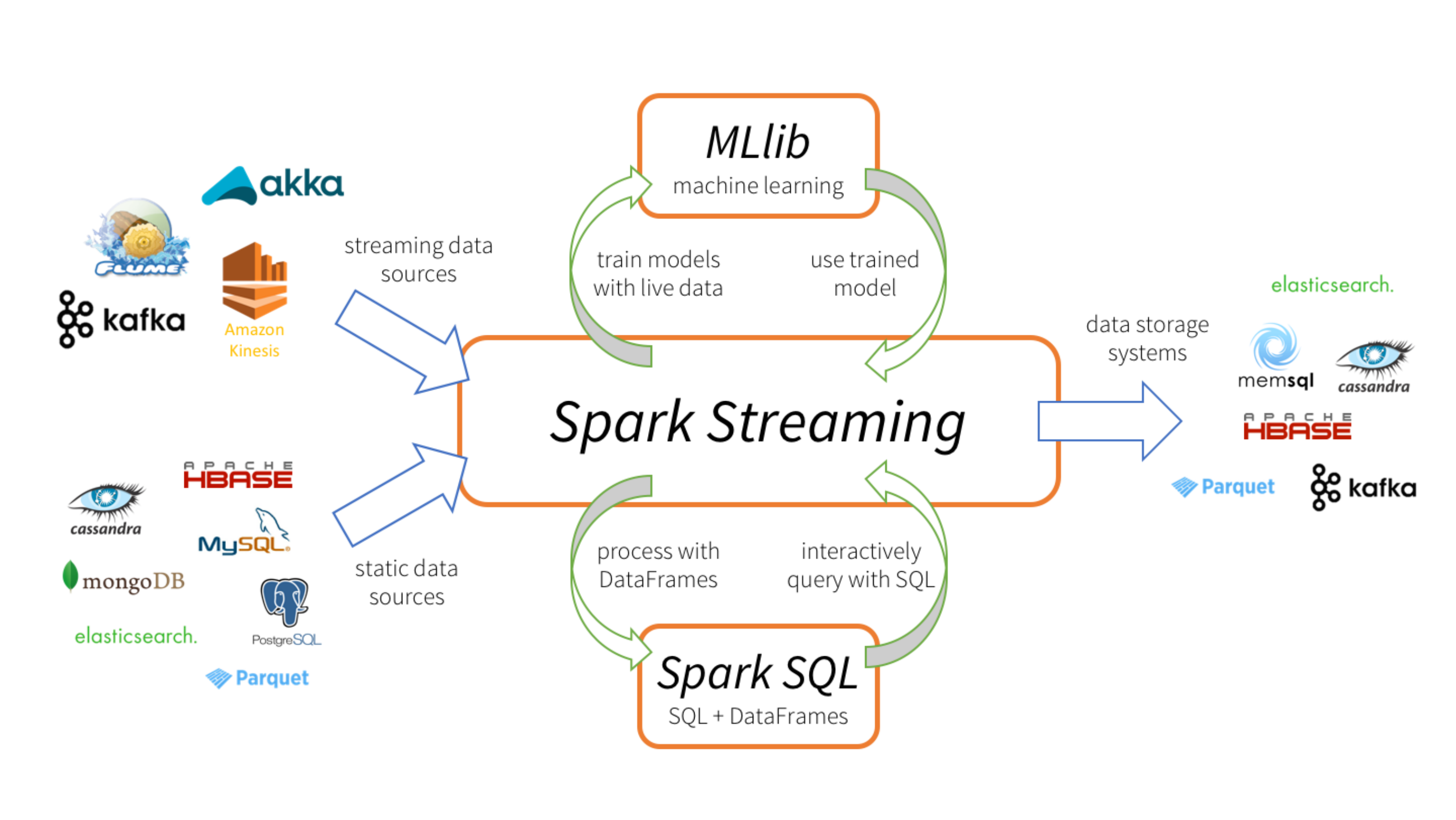
Real-Time Traffic



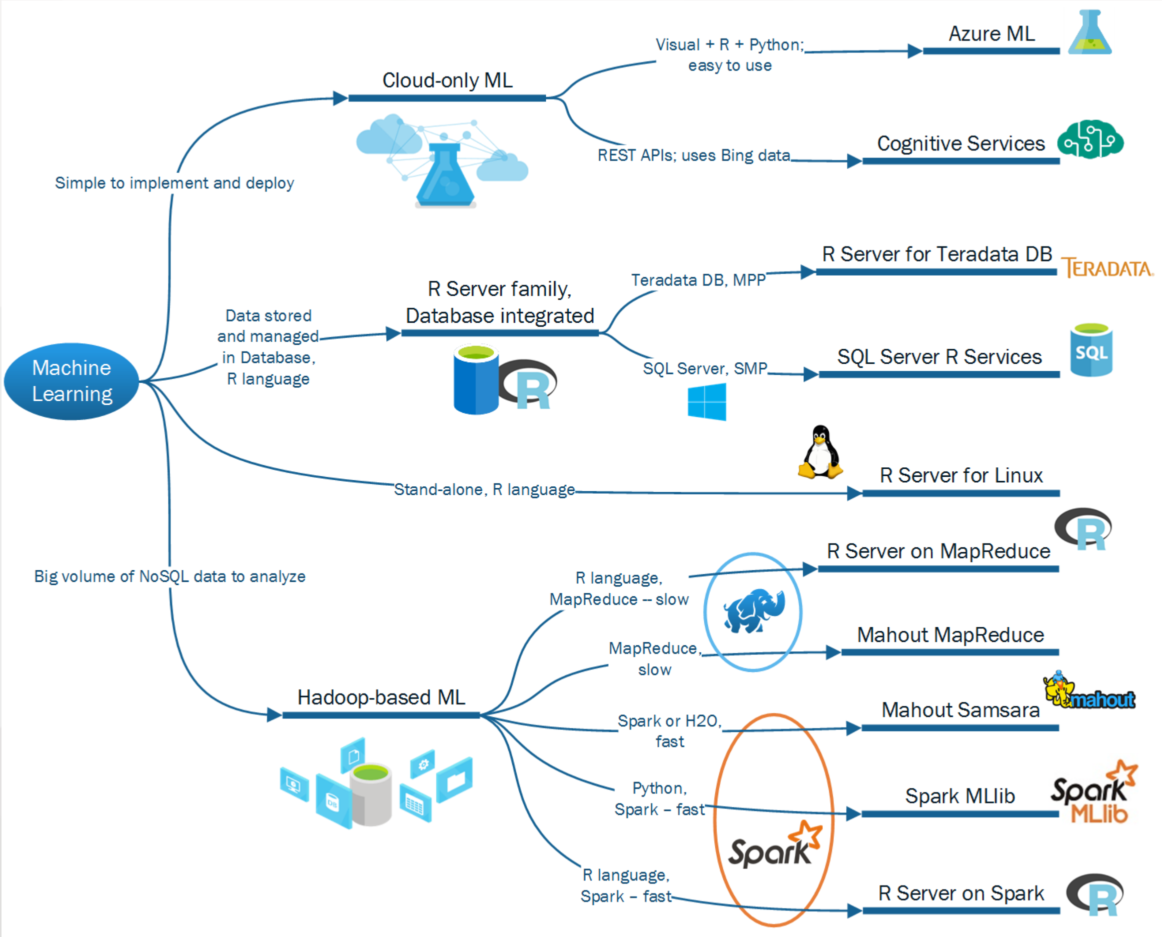
PART6

Spark Machine Learning

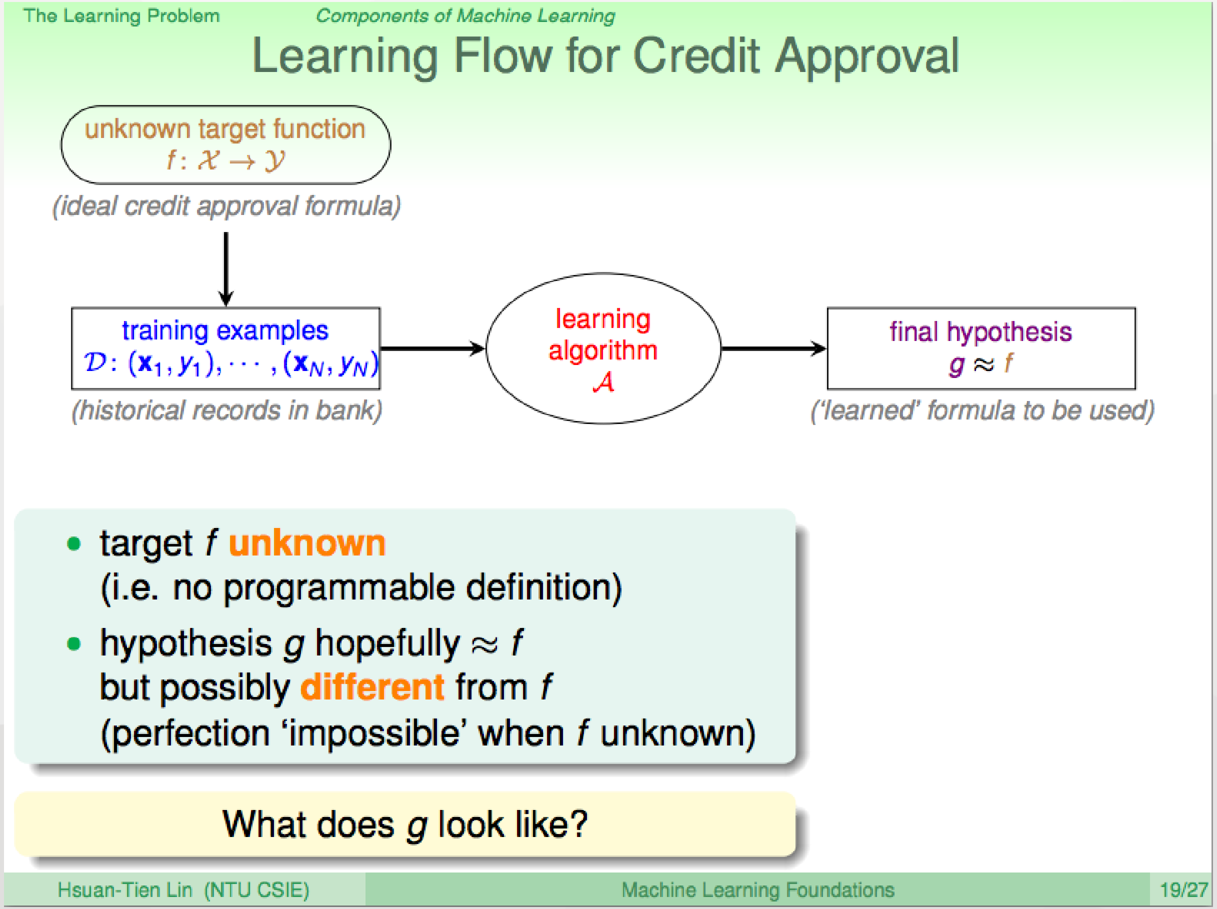
Ecosystems


akka
FLUME
kafka
Amazon Kinesis
streaming data sources
cassandra
HBASE
MySQL
mongoDB
PostgreSQL
elasticsearch.
Parquet
static data sources
MLlib
machine learning
train models with live data
use trained model
Spark Streaming
process with DataFrames
interactively query with SQL
Spark SQL
SQL + DataFrames
data storage systems
elasticsearch.
memsql
cassandra
HBASE
Parquet
kafka

Ecosystems


Machine Learning
Simple to implement and deploy
Cloud-only ML
Visual + R + Python; easy to use
Azure ML
REST APIs; uses Bing data
Cognitive Services
Data stored and managed in Database, R language
R Server family, Database integrated
Teradata DB, MPP
R Server for Teradata DB
TERADATA
SQL Server, SMP
SQL Server R Services
SQL
Stand-alone, R language
R Server for Linux
Big volume of NoSQL data to analyze
Hadoop-based ML
R language, MapReduce -- slow
R Server on MapReduce
MapReduce, slow
Mahout MapReduce
mahout
Spark or H2O, fast
Mahout Samsara
Python, Spark – fast
Spark MLlib
Spark MLlib
Spark
R language, Spark – fast
R Server on Spark

Use Case : Credit Approval



Learning Flow for Credit Approval

unknown target function
$f: \mathcal{X} \to \mathcal{Y}$

(ideal credit approval formula)

training examples
$\mathcal{D}: (\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$

(historical records in bank)

learning algorithm
$\mathcal{A}$

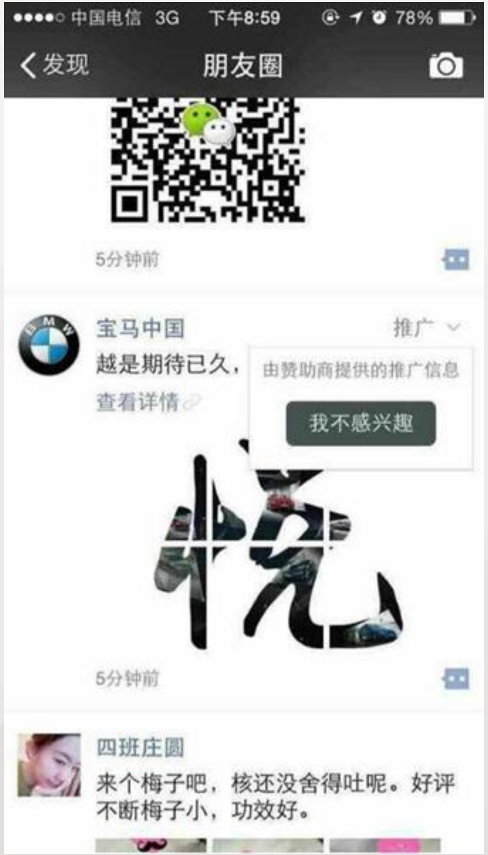
final hypothesis
$g \approx f$

('learned' formula to be used)

- target f **unknown**
  (i.e. no programmable definition)
- hypothesis g hopefully $\approx f$
  but possibly **different** from f
  (perfection 'impossible' when f unknown)

What does g look like?

Use Case : WeChat



5分钟前

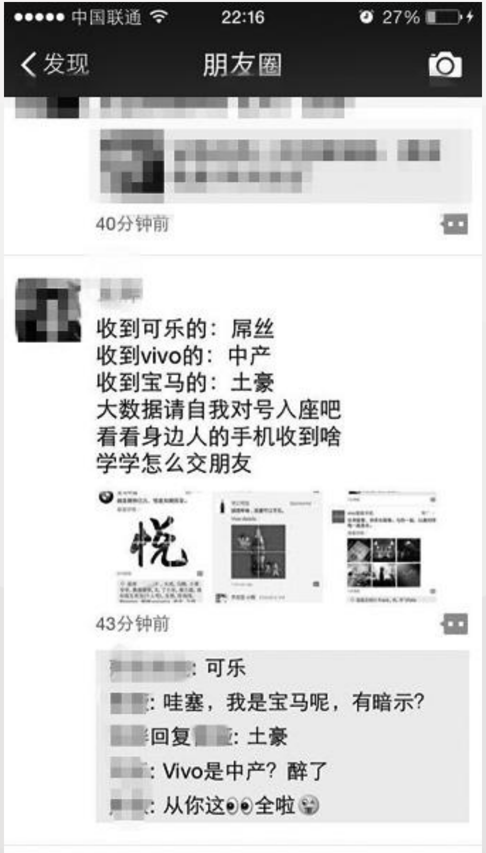
宝马中国

越是期待已久，

5分钟前

四班庄圆

来个梅子吧，核还没舍得吐呢。好评不断梅子小，功效好。



[redacted]

40分钟前

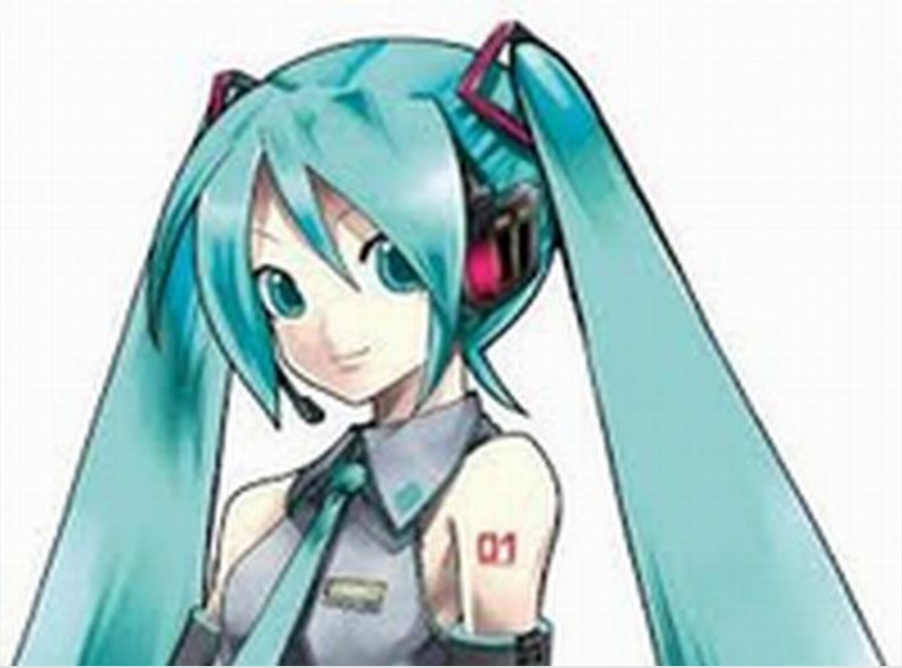
[redacted]

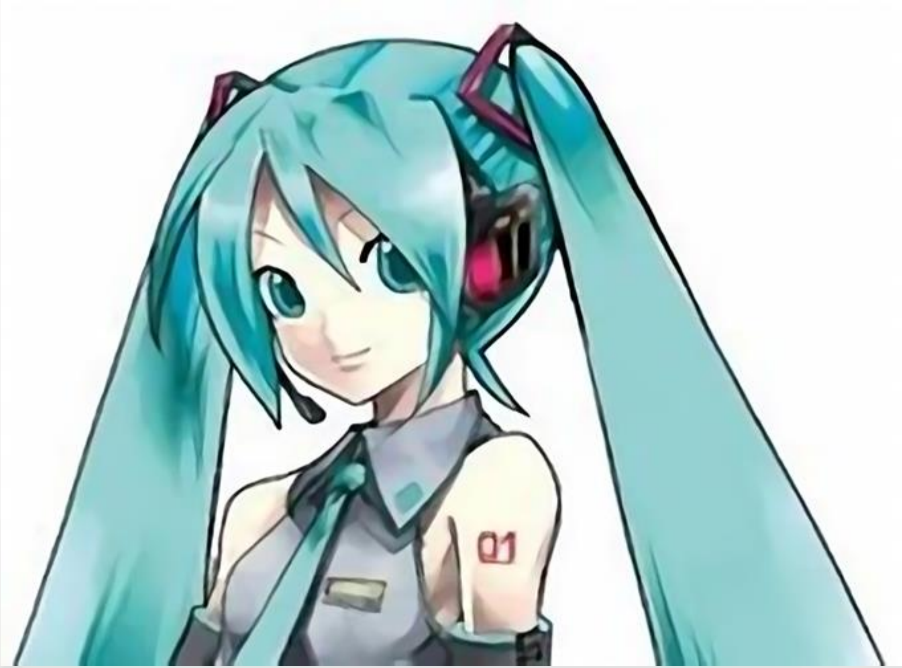
收到可乐的：屌丝
收到vivo的：中产
收到宝马的：土豪
大数据请自我对号入座吧
看看身边人的手机收到啥
学学怎么交朋友

43分钟前

[redacted]: 可乐
[redacted]: 哇塞，我是宝马呢，有暗示?
[redacted]回复[redacted]: 土豪
[redacted]: Vivo是中产? 醉了
[redacted]: 从你这👀全啦😲

Use Case : Hatsune Miku Upscale

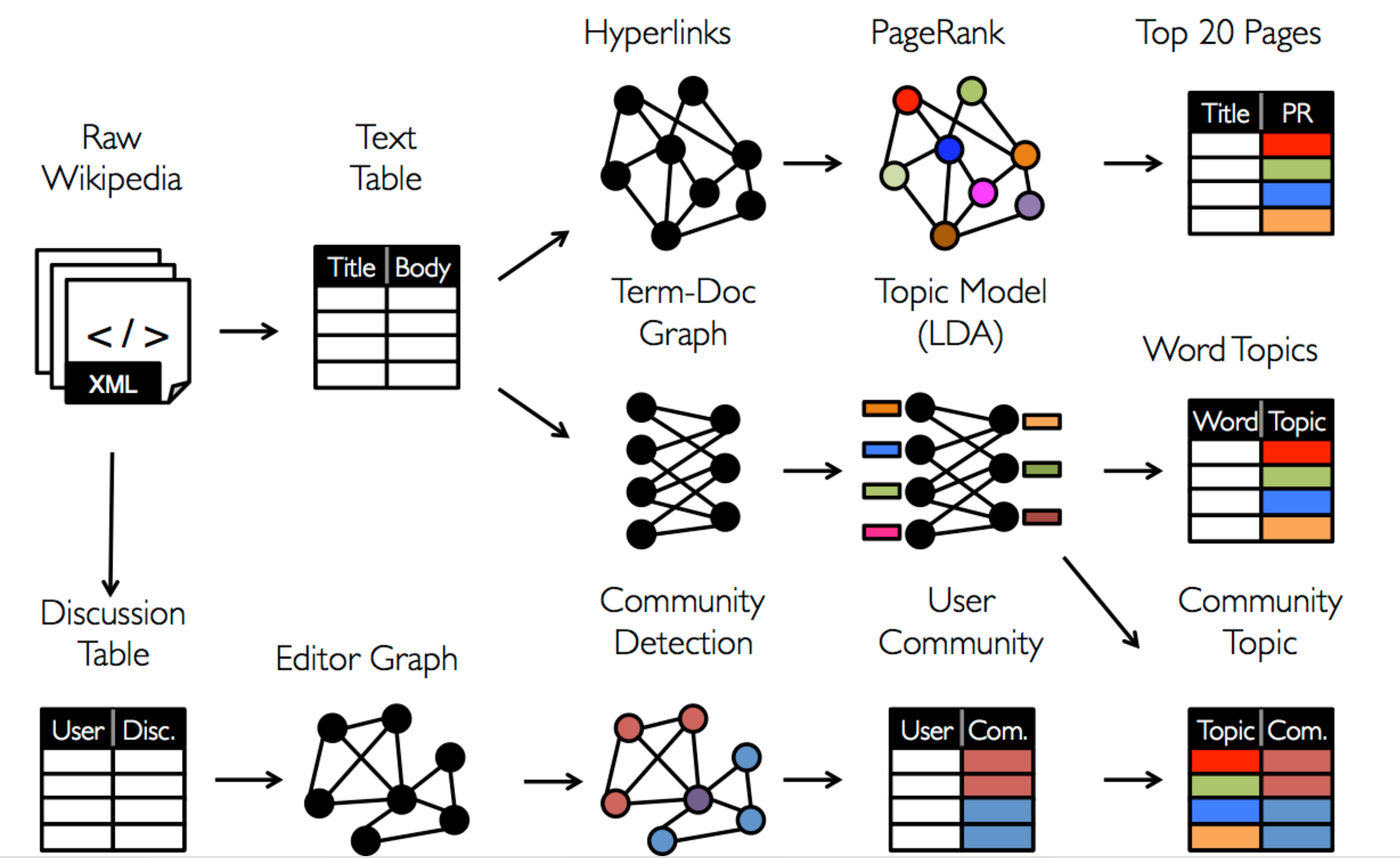




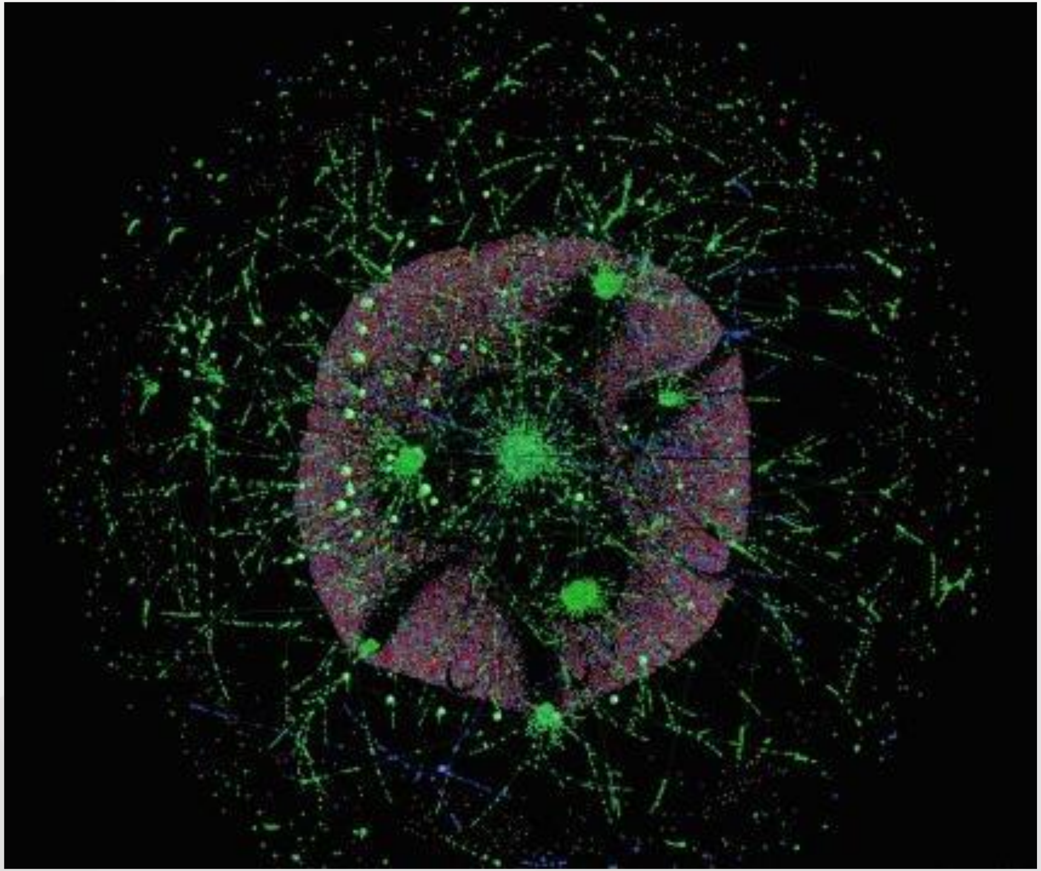
PART7

Spark GraphX

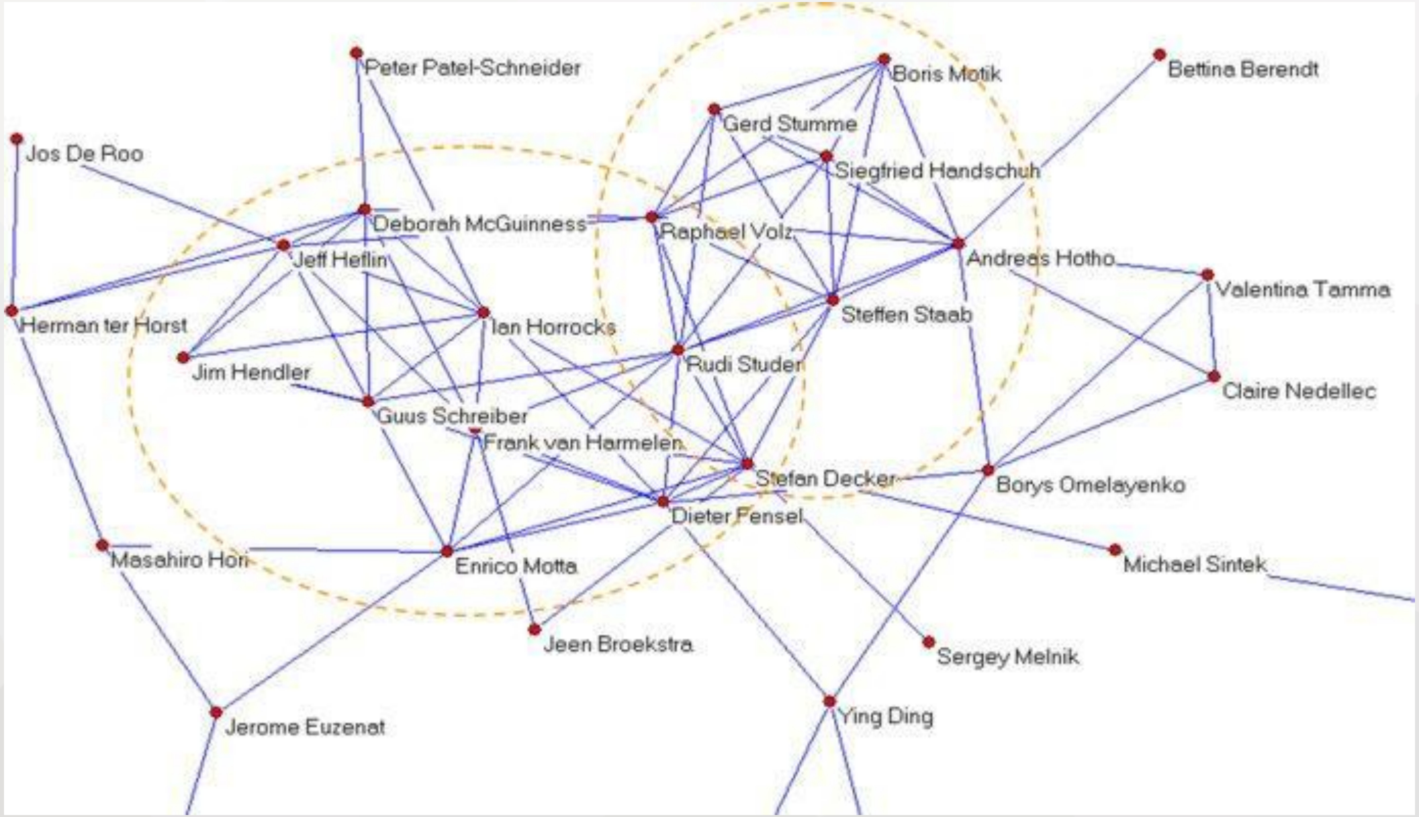
GraphX


Hyperlinks
PageRank
Top 20 Pages
Raw Wikipedia
Text Table
Title
PR
</ XML
Title
Body
Term-Doc Graph
Topic Model (LDA)
Word Topics
Word
Topic
Discussion Table
Editor Graph
Community Detection
User Community
Community Topic
User
Disc.
User
Com.
Topic
Com.

Weibo



Social Network Site community



The End