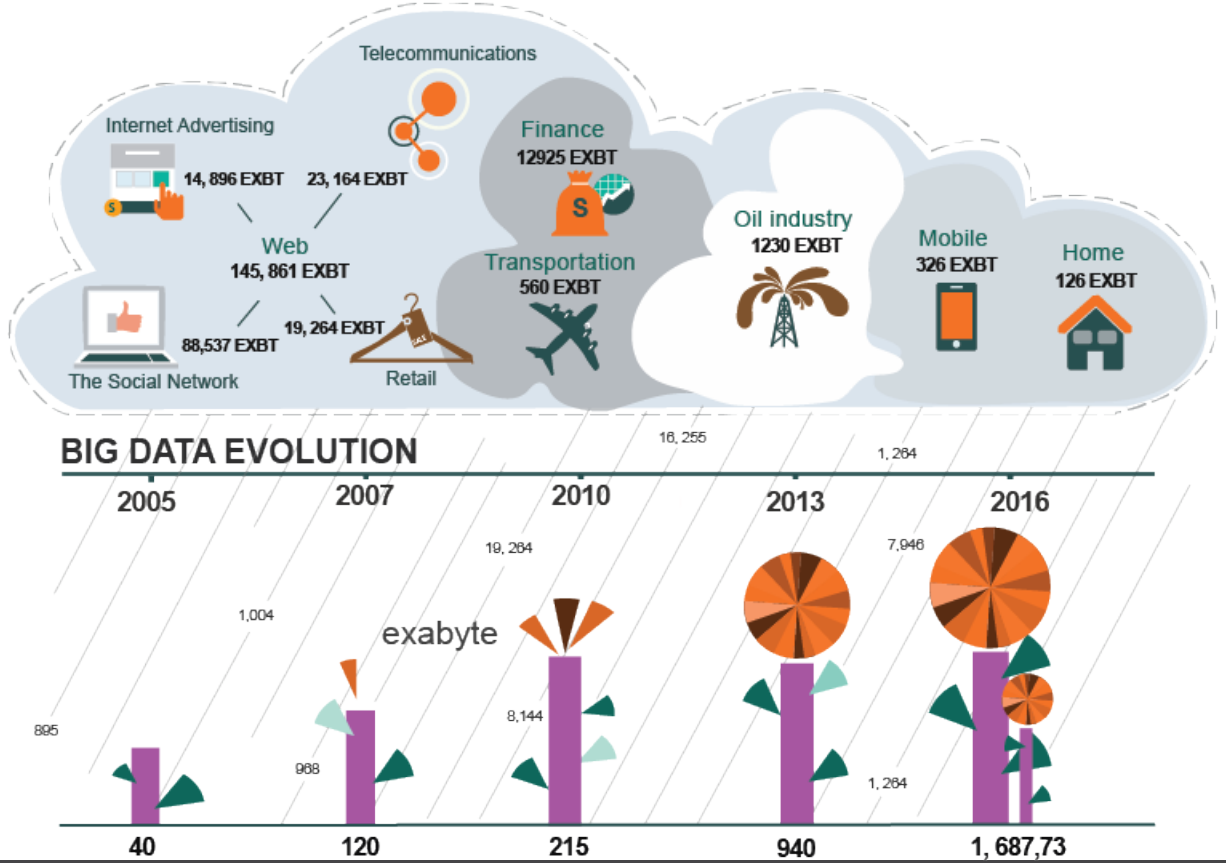
Big Data

RD Training Center Nizhniy Novgorod 2020

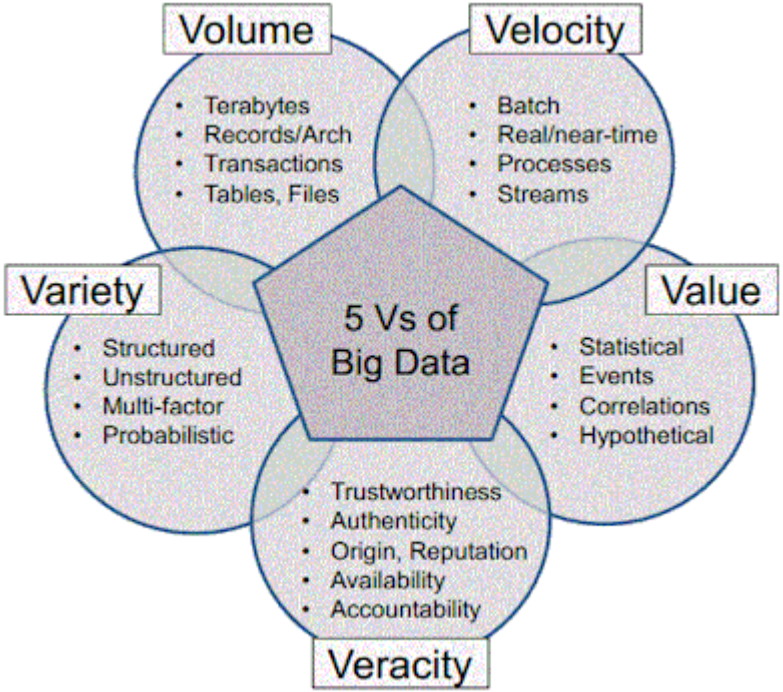
DATA AROUND US



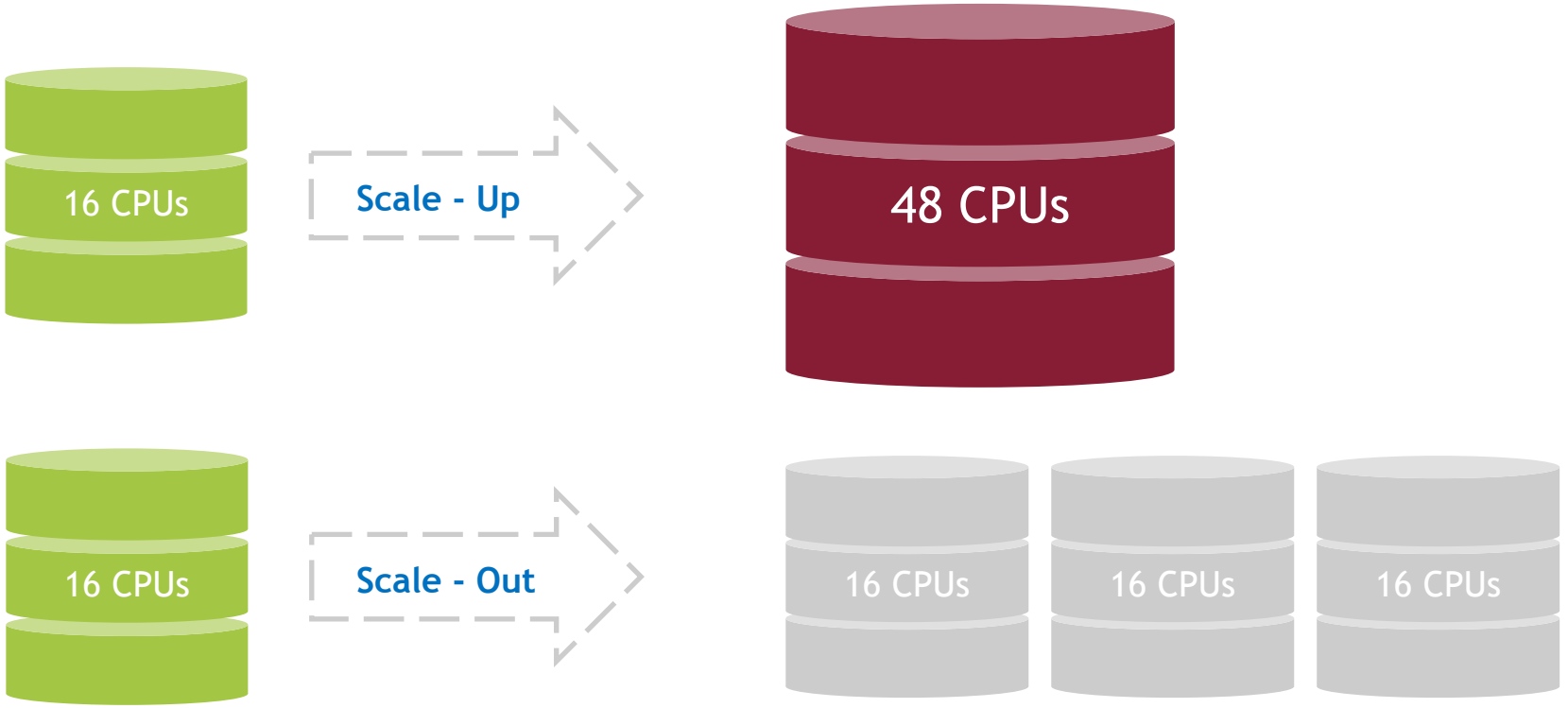
WHEN DATA BECOME “BIG”

- 3Vs (Volume + Variety + Velocity)
- 4Vs (+ Veracity)
- 5Vs (+ Value)

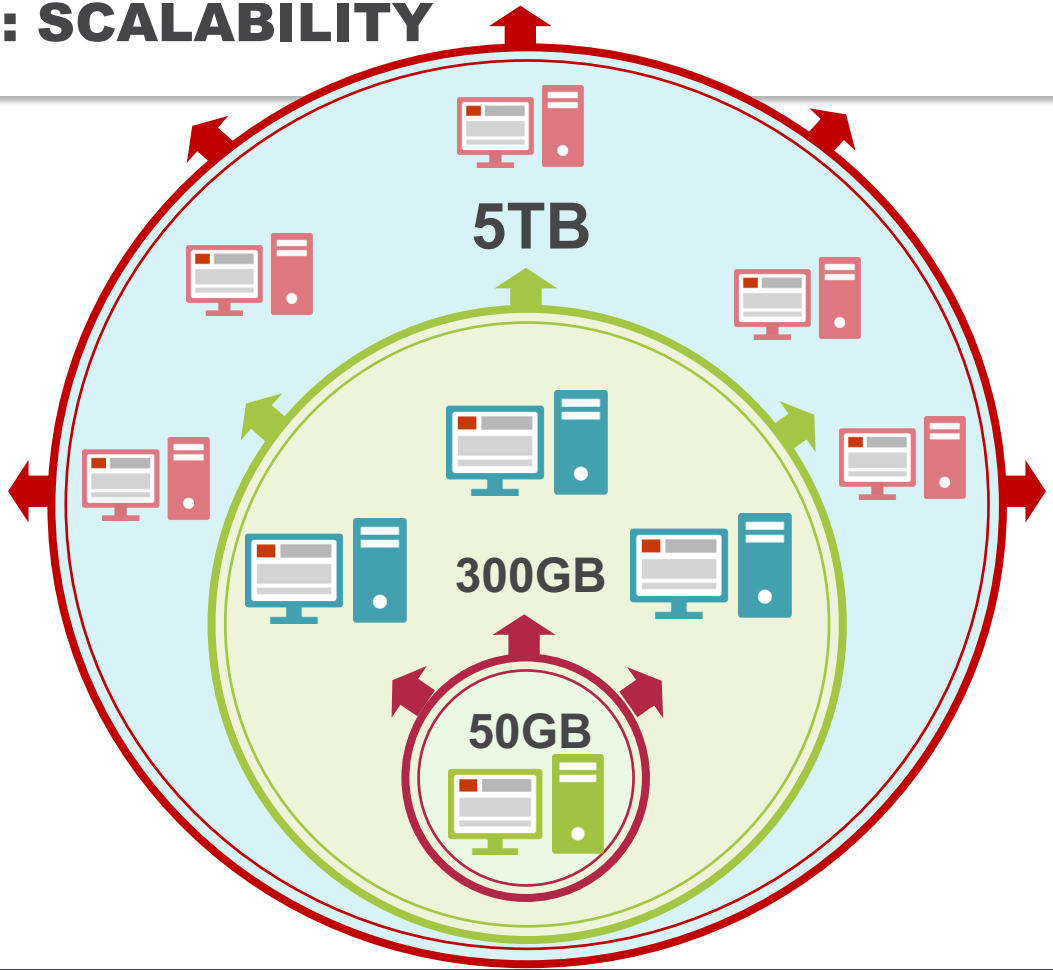
- But still… When your data become “BIG”?
  GBs? TBs? PBs?



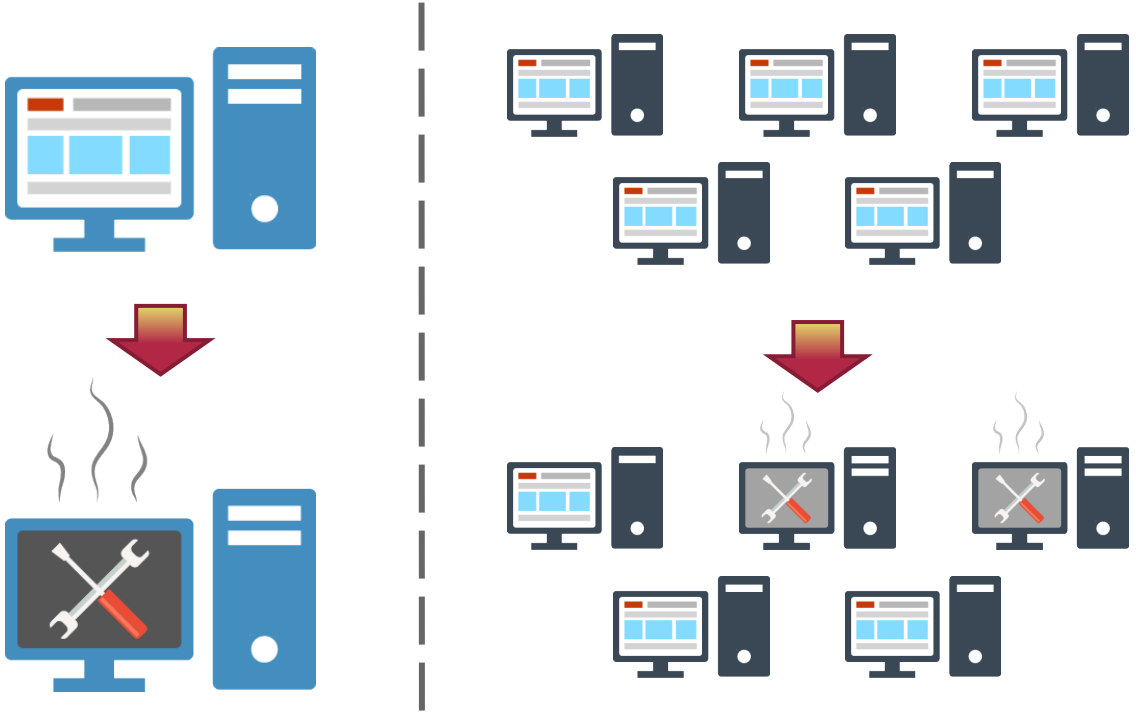
MOTIVATION: SCALE-UP VS SCALE-OUT



MOTIVATION: SCALABILITY

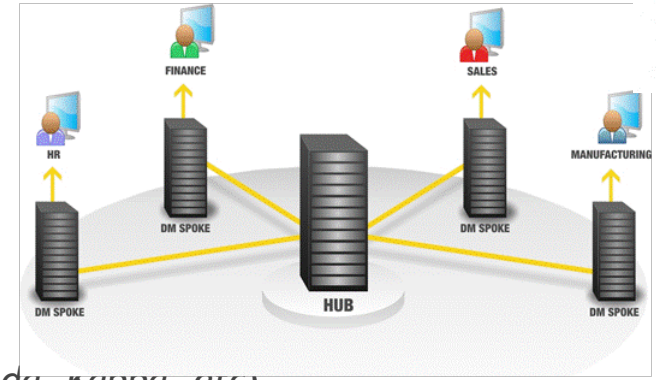


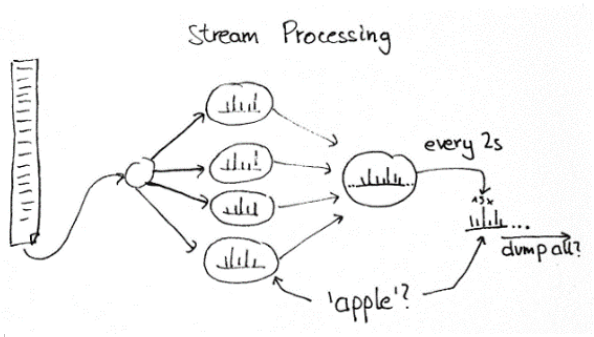
MOTIVATION: FAULT TOLERANCE



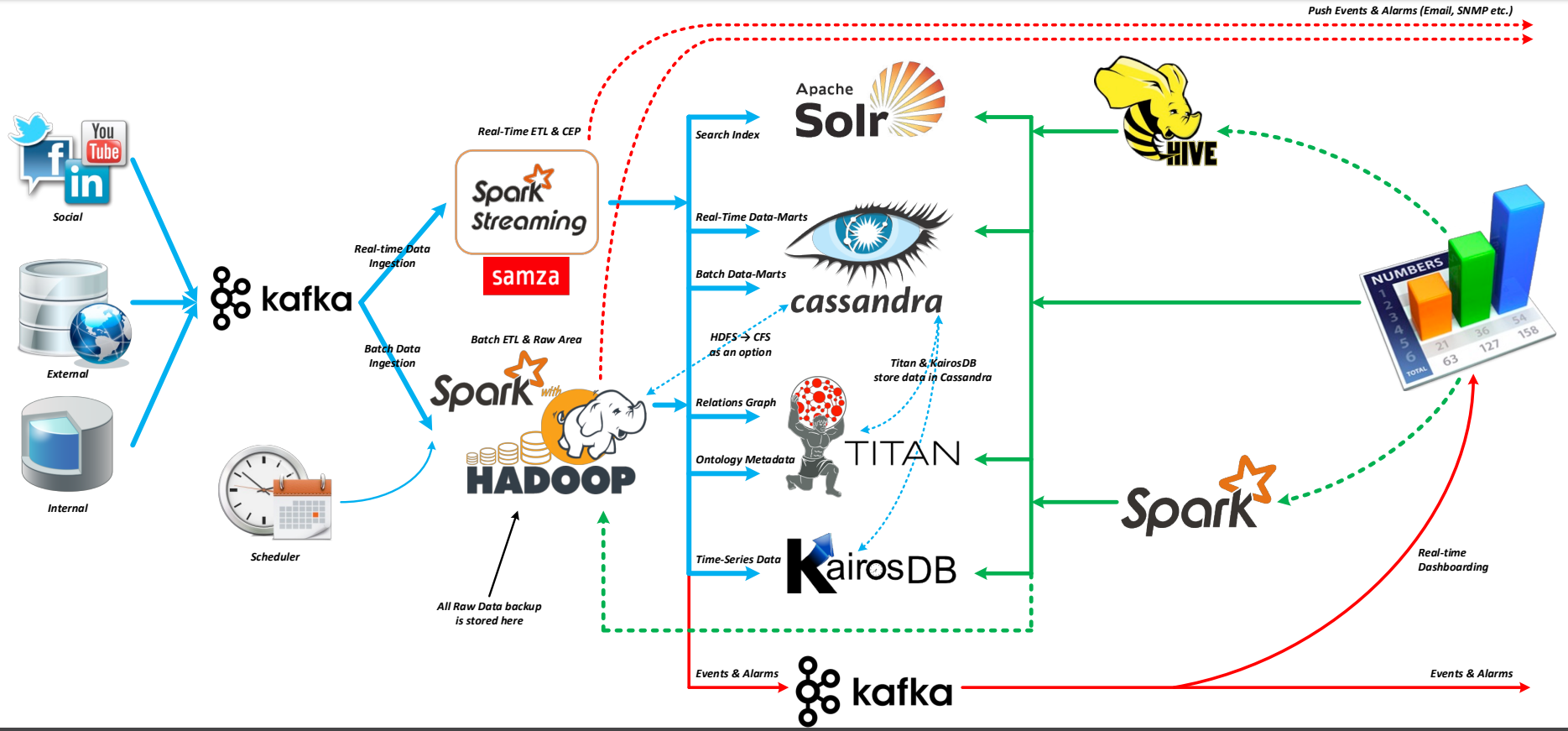
BIG DATA: IT IS NOT ONLY HADOOP

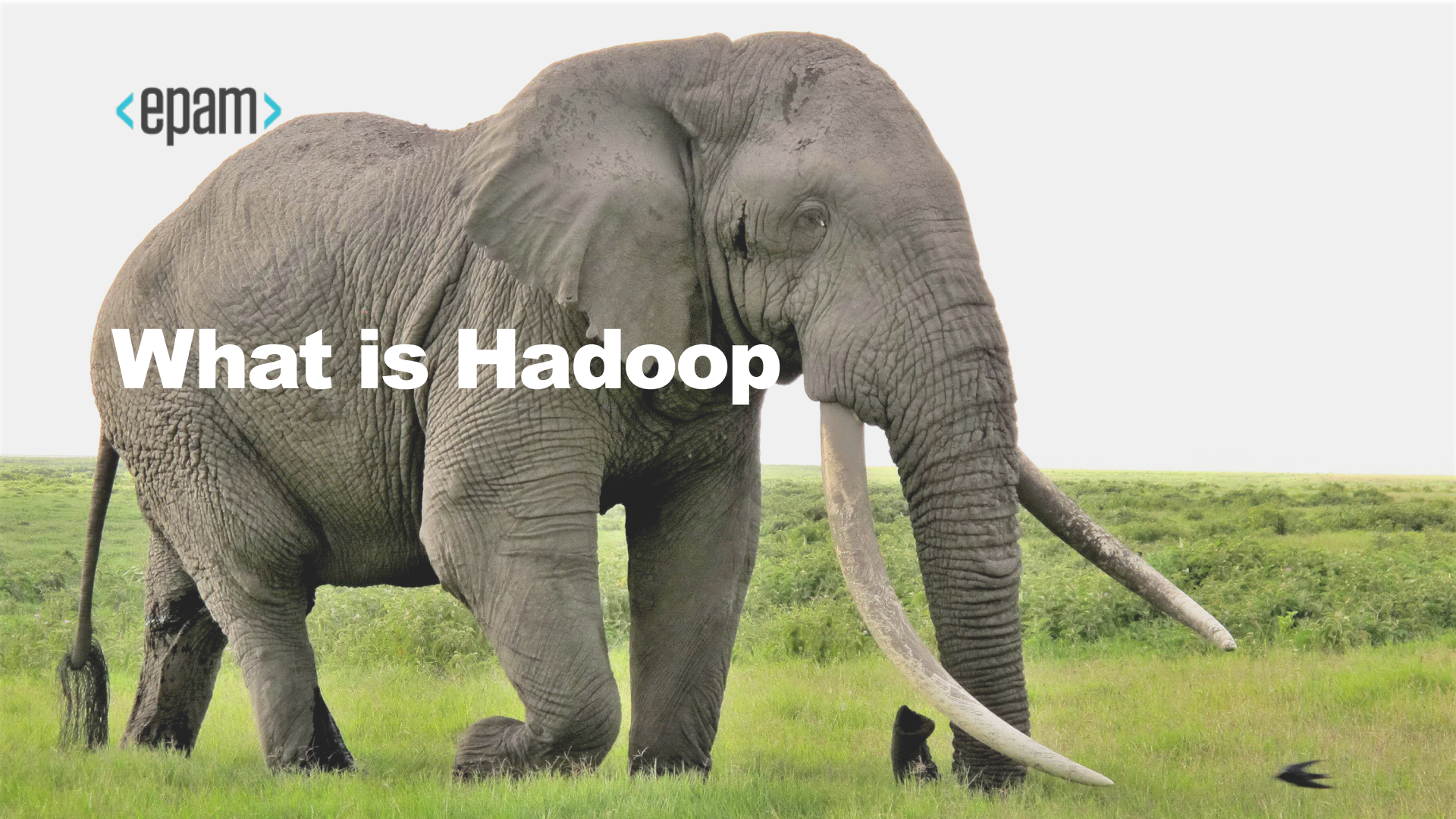
- Big Data is set of technologies: hundreds frameworks and systems!
  - Data Warehousing
  - NoSQL
  - Data Integration
  - Stream Processing
  - In-Memory Computing
  - Disaster Recovery
  - Reference architectures
    (*like Data Lake, Lambda, Kappa, etc*)





ONE OF THE LATEST “MARKETECTURES”




epam
What is Hadoop

WHAT IS HADOOP

- According to **Wikipedia**

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

  Hadoop is an open-source software framework for distributed storage and distributed processing of Big Data on clusters of commodity hardware

- According to **Apache** portal

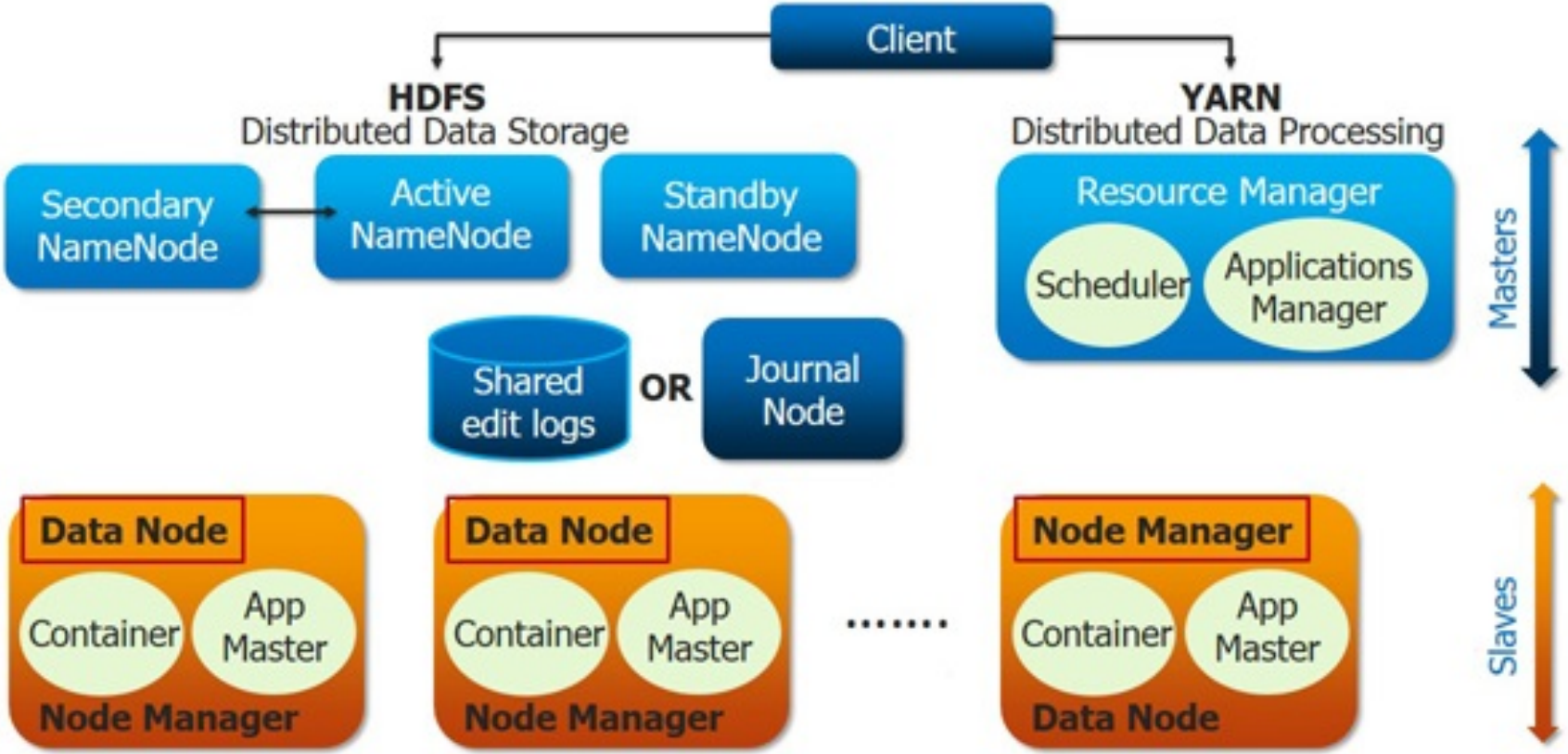
  https://hadoop.apache.org

  Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models

HADOOP: CORE COMPONENTS

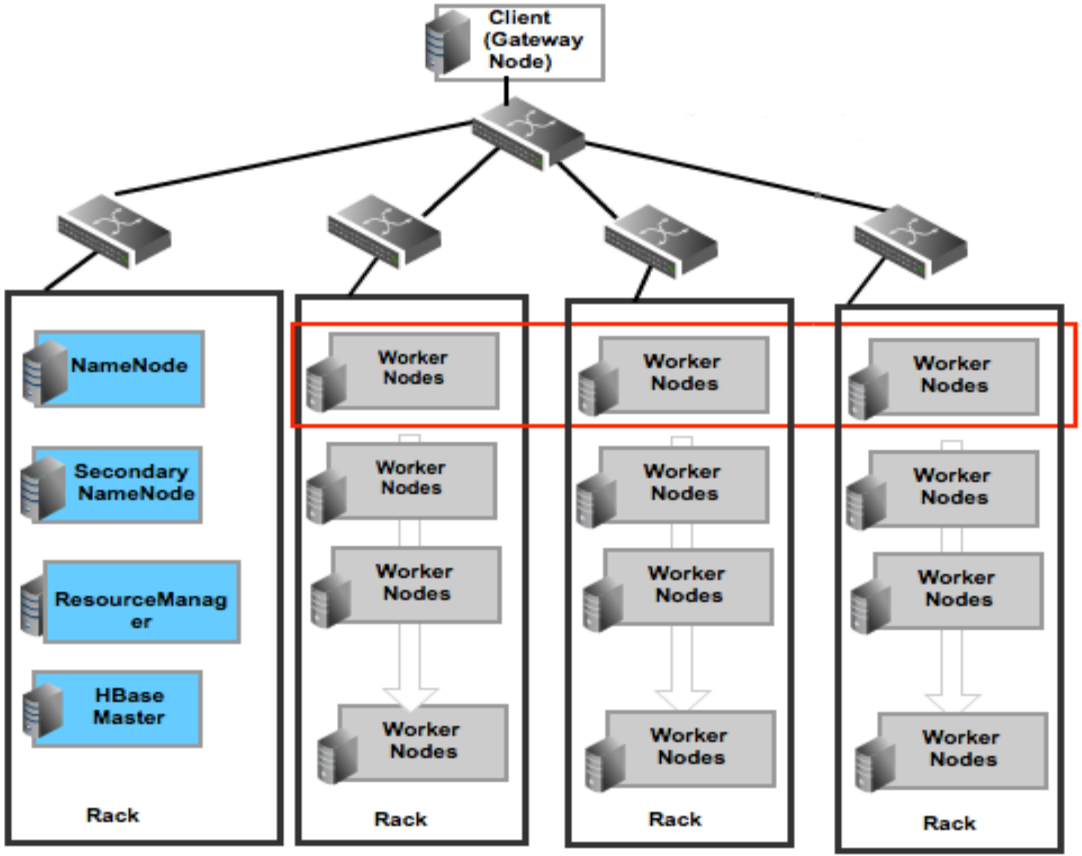
- **Hadoop Commons**
  libraries and utilities used by other Hadoop modules
- **Hadoop Clients**
  libraries and utilities used to access Hadoop's components
- **HDFS – Hadoop Distributed File System**
  scalable system that stores data across multiple machines without prior organization
- **Yarn – Yet Another Resource Negotiator**
  resource management framework for scheduling and handling resource requests from distributed apps
- **MapReduce**
  software programming model for processing large sets of data in parallel
  in fact, distributed application on top of Yarn
- **Spark**
  open-source cluster-computing framework designed for speed and ease of use.

HADOOP: ARCHITECTURE



HADOOP: TYPICAL TOPOLOGY

- DataNodes, NodeManagers and RegionServers are typically co-deployed, to follow Data Locality strategy
- Best practice is to have at least 3 replicas of data located in separate racks
- Best practice is to add DataNodes according replication-factor into separate racks




epam
HDFS

WHAT IS HDFS

- HDFS – Hadoop Distributed File System
- HDFS is scalable system written in Java that stores data across multiple machines without prior organization

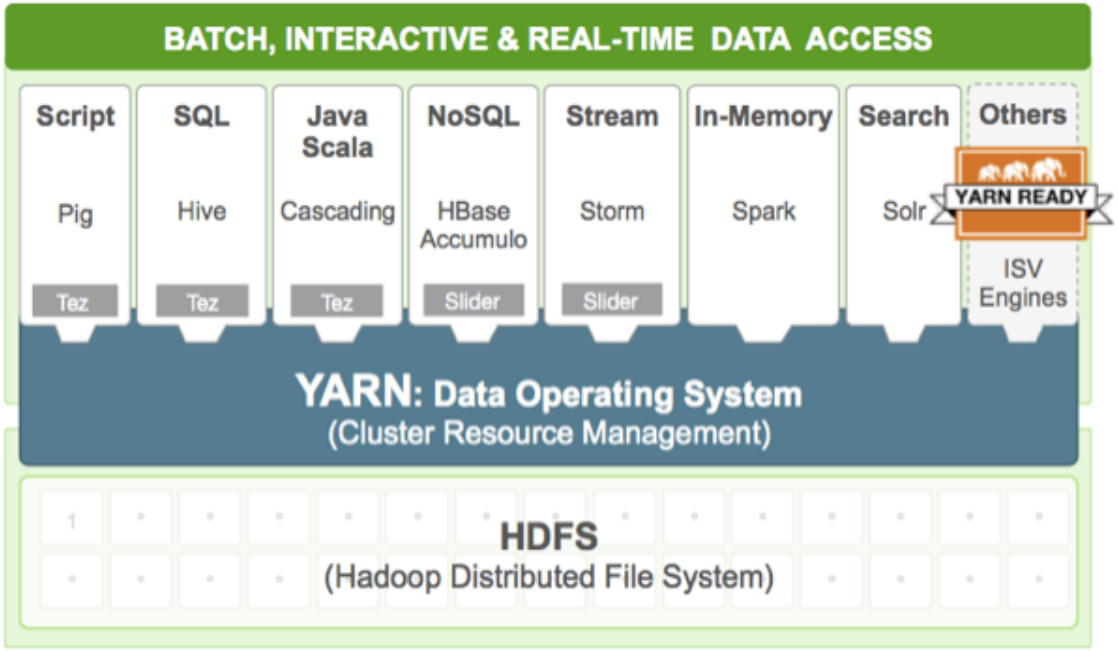
- According to Hortonworks:
  HDFS has demonstrated production scalability of up to 200PB of storage and a single cluster of 4500 servers, supporting close to a billion files and blocks

- HDFS provides
  - Distributed data storage
  - Reliable and secure operations with data
  - Scalability and efficiency
  - Fault tolerance and high availability
  - Coordination with Yarn


epam
Yarn

WHAT IS YARN

- Yarn – Yet Another Resource Negotiator
- Yarn is the architectural center of Hadoop
- Yarn is unified resource manager
- It manages and spreads all resources across a cluster
- MapReduce is just one of Yarn-based applications, which “asks” Yarn for some resources
- Beside of MR, a lot of other applications are able to receive benefits from Hadoop's distributed resources




epam
Spark

WHAT IS SPARK

- Spark is a fast growing and general engine for large-scale data processing
- Up to 100x faster than MapReduce
- Runs on Hadoop, Mesos, standalone, or in the cloud
- Support for many programming languages
- SQL, streaming, and complex analytics
- Multiple options and libraries

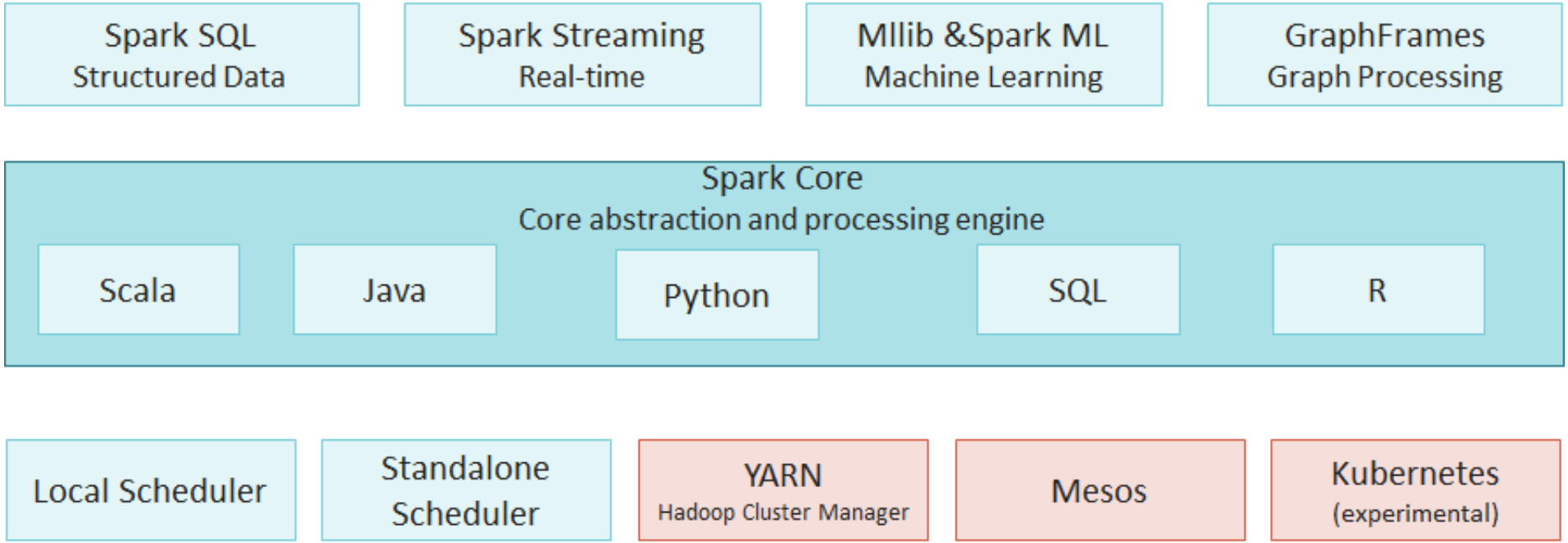
SPARK VS MAPREDUCE…

	Hadoop MapReduce	Apache Spark
Coding	complex, lengthy	compact
Coding complexity	difficult	easy
Interactive mode	no	yes
Developed in	Java	Scala
Language support	Java, C/C++, Ruby, Python, …	Scala, Java, Python, R, SQL
OS support	Linux, Windows, Max OS	Linux, Windows, Mac OS
Latency	disk oriented	memory oriented
Performance	lower	faster
Hardware	commodity	mid to high-level

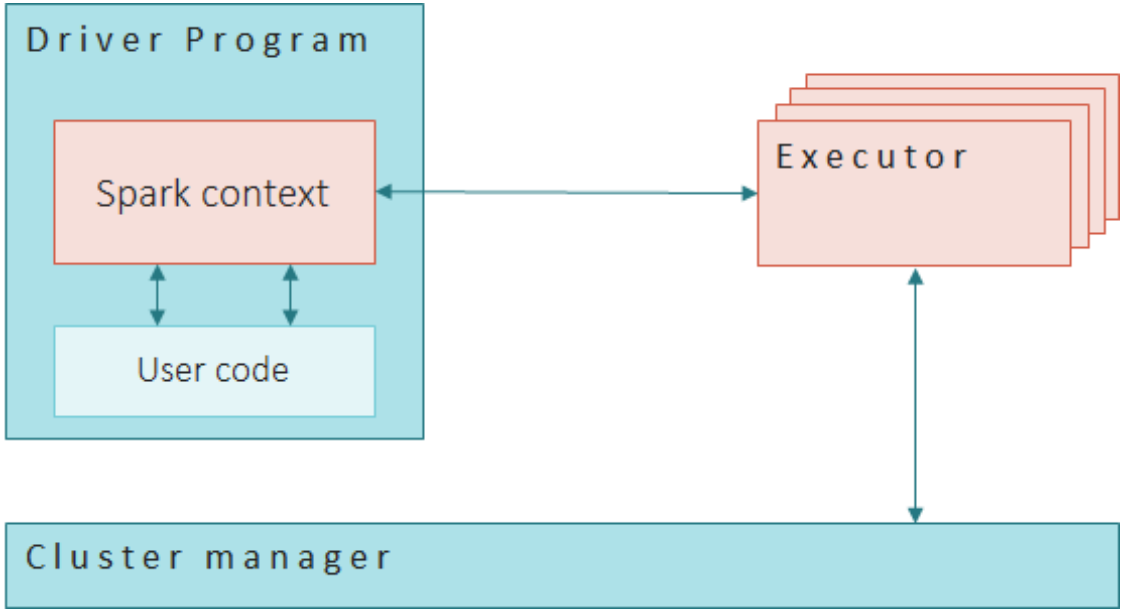
…SPARK VS MAPREDUCE

	Hadoop MapReduce	Apache Spark
Category	data processing engine	data analytics engine
Data processing	Batch	batch, streaming
Flow scheduler	3rd party (e.g. Oozie)	built-in
Fault tolerance	replication	RDD
Recovery	resilient	DAG / checkpoints
Security	Kerberos / ACLs	Shared secret password auth
Scalability	high	high
Machine learning	3rd party (e.g. Apache Mahout)	Spark ML, MLlib

SPARK: COMPONENTS

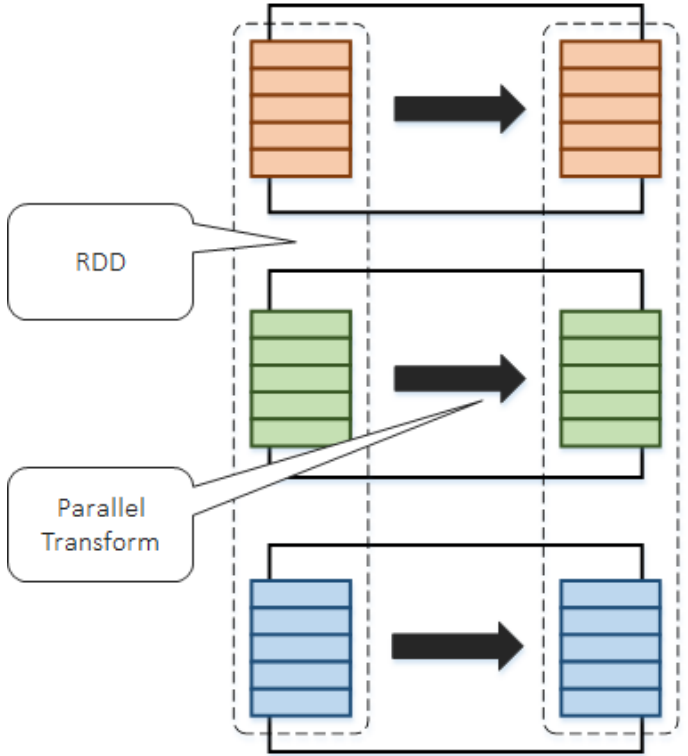


SPARK: APPLICATION

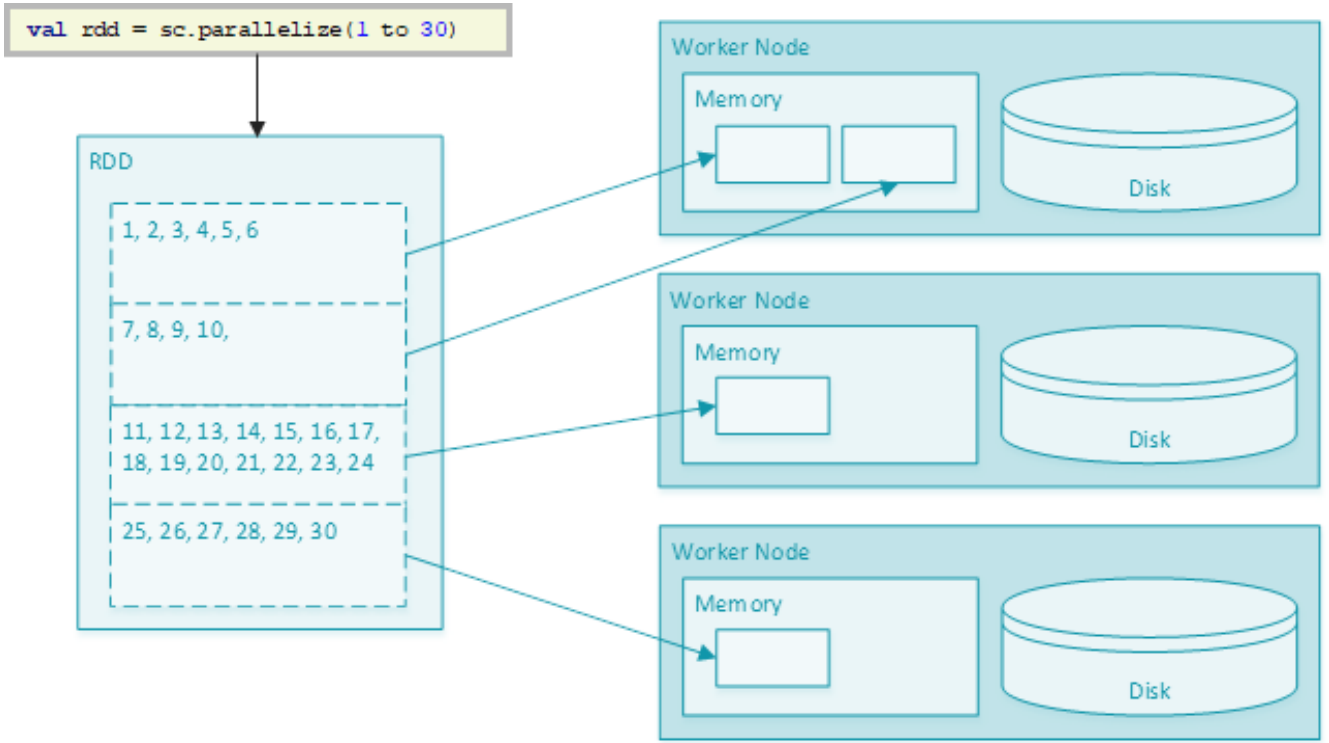


SPARK: RDD

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.



SPARK: PARTITIONING




Clouds

AWS: ARCHITECTURE


S3


Glue


Athena


Lambda


Cloudformation


Cognito


Developer tools

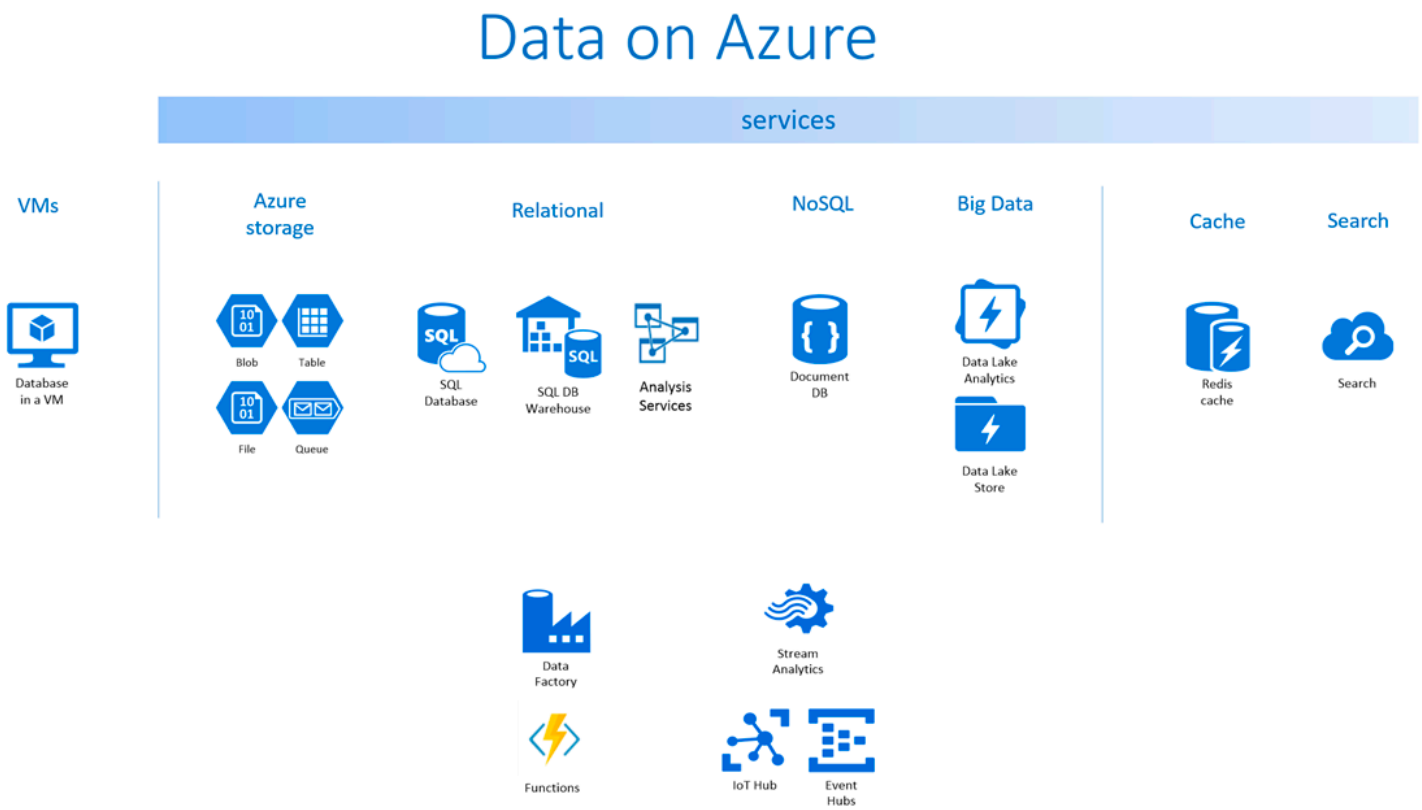

CodeCommit


CodeDeploy


CodeBuild


CodePipeline

AZURE: ARCHITECTURE



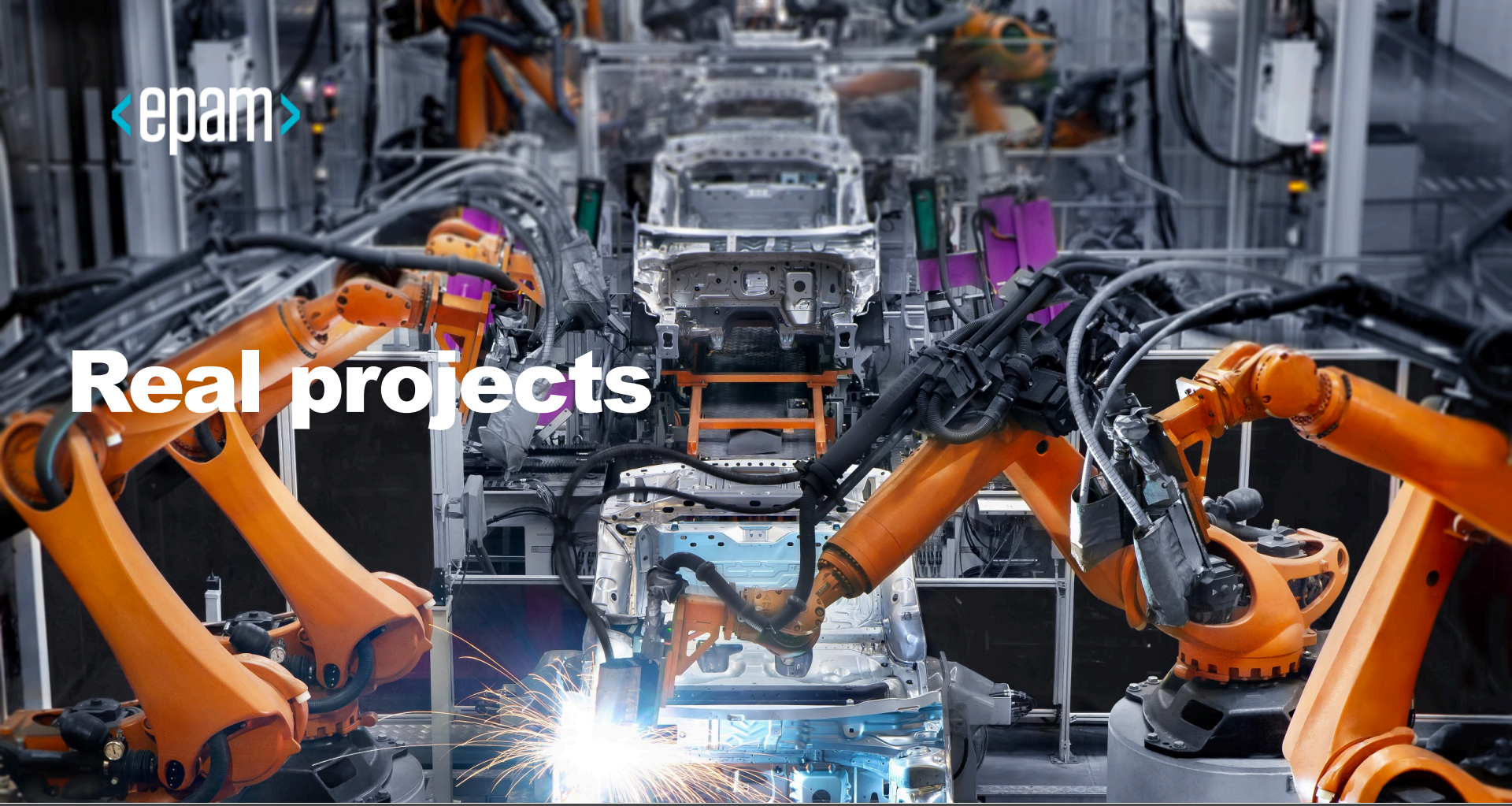
EPAM CLOUD

THE **MAIN AIM** OF EPAM CLOUD IS **TO PROVIDE** NECESSARY **DEVELOPMENT ENVIRONMENT** FOR EPAM ENGINEERS

EPAM CLOUD IS A **SELF-SERVICE**

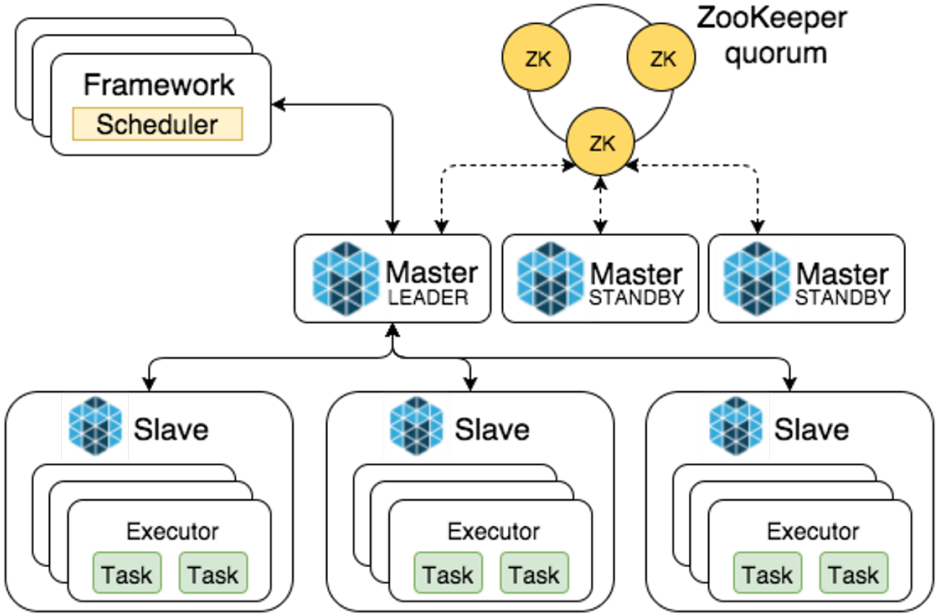
EPAM CLOUD MANAGEMENT TOOLS:

- Maestro CLI
- Web-interface


epam
Real projects

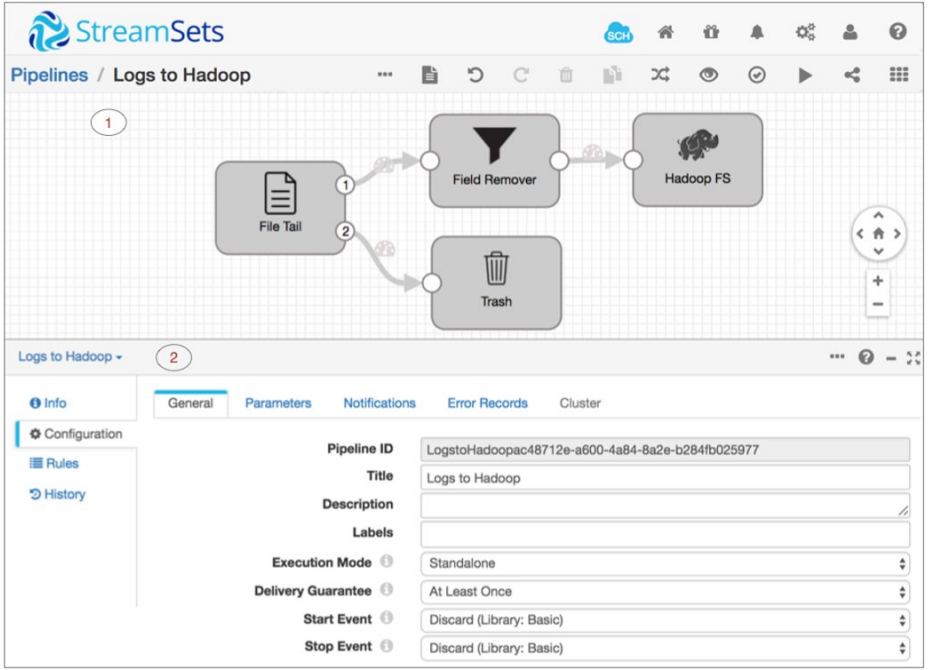
APACHE MESOS

Apache Mesos is the distributed systems kernel that abstracts the entire datacenter into a single pool of computing resources.



STREAMSETS: ARCHITECTURE

- Found by ex-Cloudera engineering leader and ex-Informatica product leader in 2014
- Visual ETL tool
- The core software is developed in Java, the web
  interface is developed in Javascript/Angular JS, D3.js,
  HTML and CSS
- 100% open source, based on the Apache 2.0 license
- No need to learn programming to create pipelines



STREAMSETS: LANGUAGE

USE CASES

- To configure expressions and conditions in processors, such as the Expression Evaluator or Stream Selector
- To define any stage or pipeline property that represents a numeric or string value

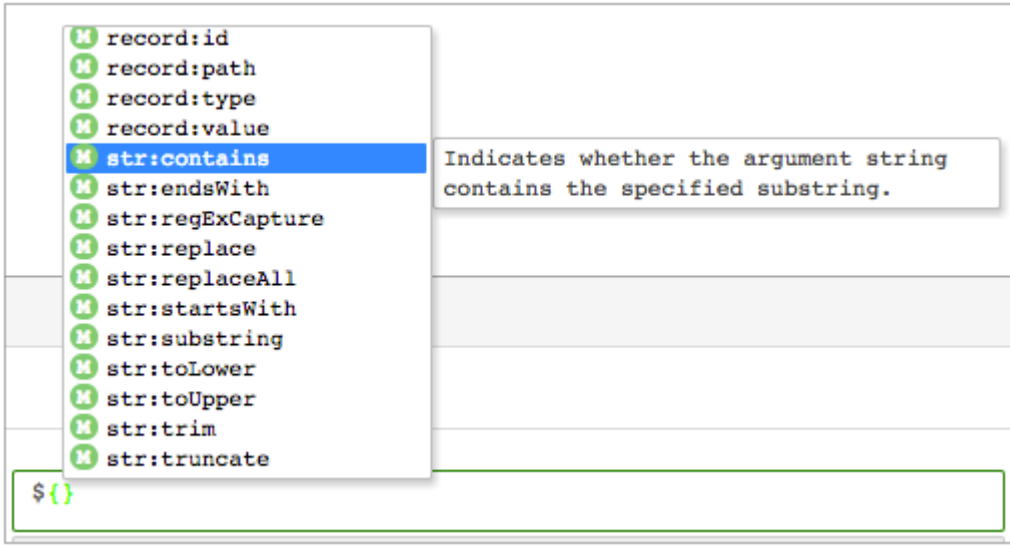
ELEMENTS:

- Constant, Datetime variables, Field names, Functions, Literals, Operators, Runtime Value

EXPRESSION COMPLETION

EXAMPLES:

- ${record:value('/payment_type') == 'CRD'}
- ${record:type('/ID')=STRING}
- ${str:toUpper(record:value('/STATE')}
- ${record:value('/total_amount')
  - (record:value('/tip_amount')
  + record:value('/tolls')
  + record:value('/mta_tax'))}



mongoDB.

Create

- db.collection.insertOne()
- db.collection.insertMany()

```
db.users.insertOne(                 <- collection
   {
      name: "sue",                  <- field: value
      age: 26,                      <- field: value
      status: "pending"             <- field: value
   }                                } document
)
```

Read

- db.collection.find()

```
db.users.find(                      <- collection
   { age: { $gt: 18 } },            <- query criteria
   { name: 1, address: 1 }          <- projection
).limit(5)                          <- cursor modifier
```

Update

- db.collection.updateOne()
- db.collection.updateMany()
- db.collection.replaceOne()

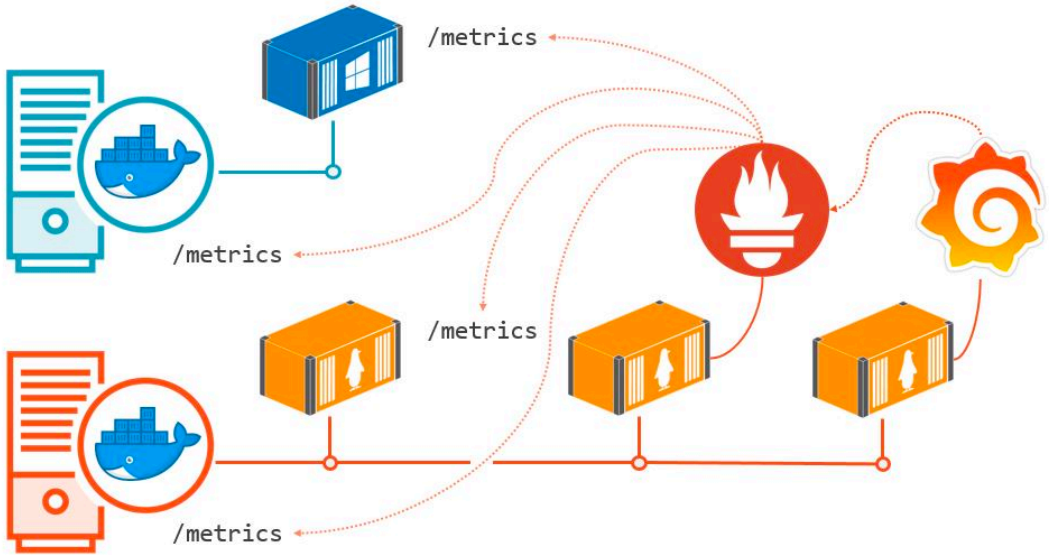
```
db.users.updateMany(                <- collection
   { age: { $lt: 18 } },            <- update filter
   { $set: { status: "reject" } }   <- update action
)
```

Delete

- db.collection.deleteOne()
- db.collection.deleteMany()

```
db.users.deleteMany(                <- collection
   { status: "reject" }             <- delete filter
)
```

PROMETHEUS



- Timeseries DB and monitoring system
- Written in Go
- High performance, low cpu usage, good data compaction, small memory footprint
- Pull model
- Official libraries for most popular languages
- God knows how many unofficial libraries
- Hundreds of available exporters. If you use some service, there is most likely an exporter for it.
- Exporters!

First Job

SPARK: WORDCOUNT – INPUT AND OUTPUT

Sample input:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Expected output:

Duis	1
Excepteur	1
Lorem	1
Ut	1
ad	1
adipisicing	1
aliqua.	1
aliquip	1
amet,	1
anim	1
aute	1
cillum	1
commodo	1
consectetur	1
consequat.	1
culpa	1
cupidatat	1 … … …

MAPREDUCE

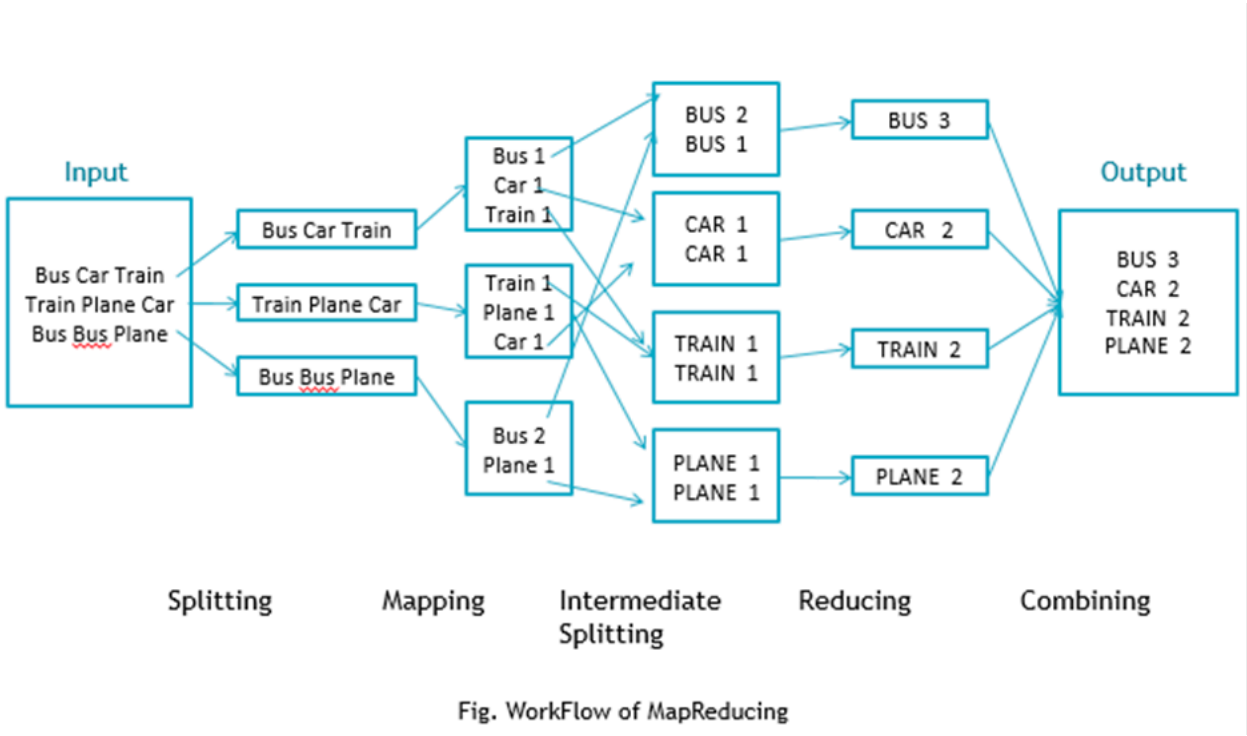
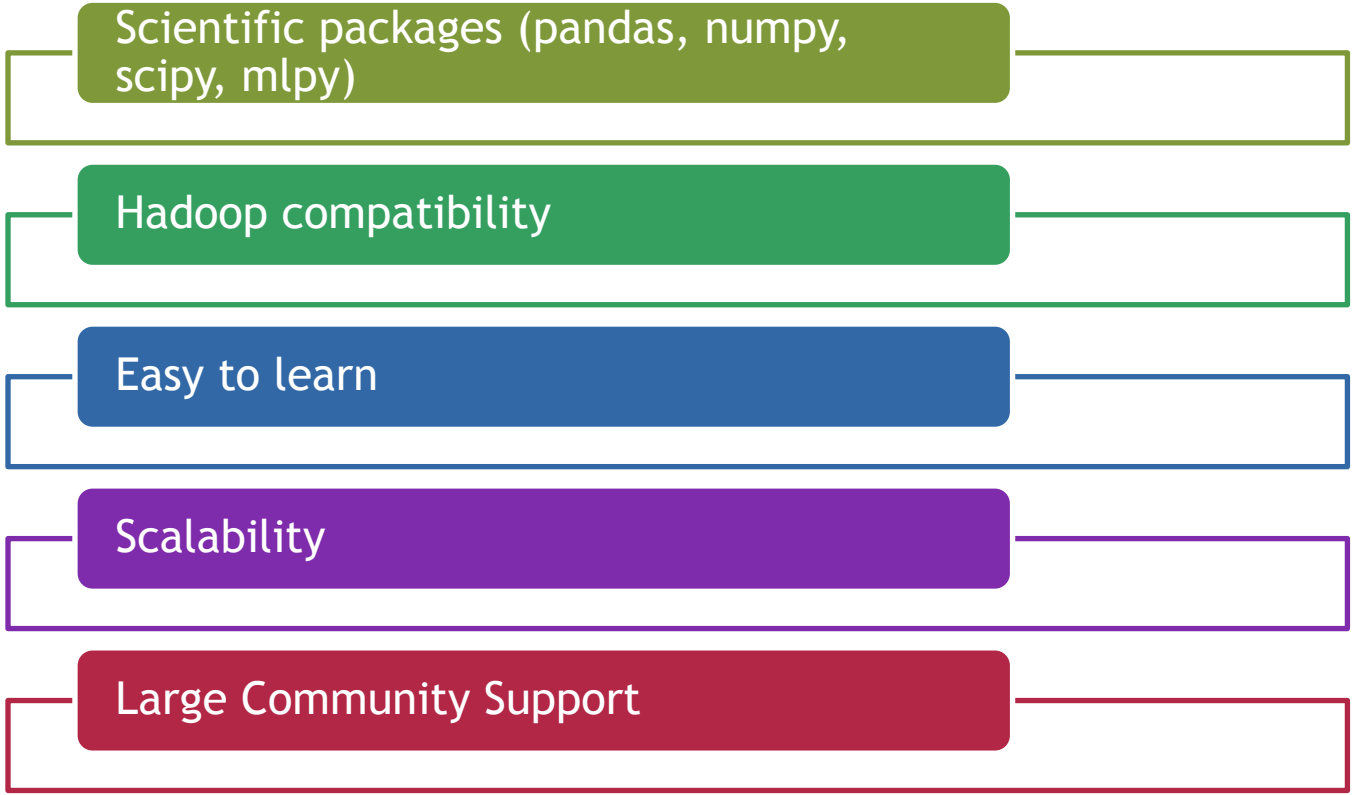


Fig. WorkFlow of MapReducing

SPARK: WORDCOUNT – CODE

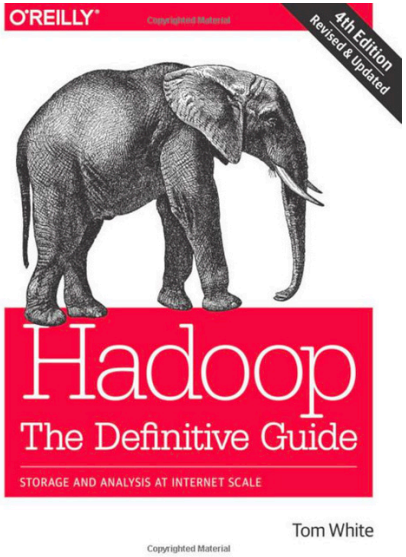
```
from pyspark.sql import SparkSession
from pyspark.conf import SparkConf

conf=SparkConf()\
    .setMaster(“local[*]”)\
    .setAppName(“WordCount”)\
    .setExecutorEnv(“spark.executor.memory”, ”4g”)\
    .setExecutorEnv(“spark.driver.memory”, ”4g”)

spark=SparkSession.builder\
    .config(conf=conf)\
    .getOrCreate()

lines_rdd=spark.textFile(“~/Python/rdl/lorem_ipsum.txt”)
lines_rdd= lines_rdd.flatMap(lambda line: line.split(“ “))
lines_rdd=lines.rdd.filter(lambda x:x!=“”)
lines_count=lines_rdd.map(lambda word:(word, 1))
lines_count_result=lines_count.reduceByKey(lambda x,y:(x+y)).sortByKey()
```

WHY PYTHON IS PERFECT FOR BIG DATA



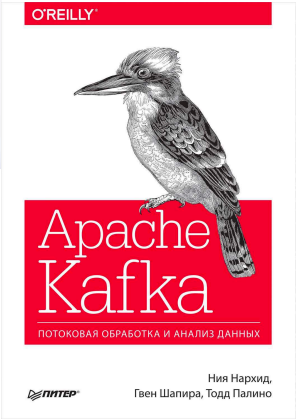
Books

LITERATURE


O'REILLY
Copyrighted Material
4th Edition
Revised & Updated
Hadoop
The Definitive Guide
STORAGE AND ANALYSIS AT INTERNET SCALE
Tom White
Copyrighted Material


Изучаем
Spark
Молниеносный анализ данных
Холден Карау, Энди Конвински,
Патрик Венделл, Матей Захария
O'REILLY


Конвейер реального времени
Потоковая обработка данных
Эндрю Дж. Псаллис
MANNING


O'REILLY
Apache Kafka
ПОТОКОВАЯ ОБРАБОТКА И АНАЛИЗ ДАННЫХ
Ния Нархид,
Гвен Шапира, Тодд Палино
ПИТЕР

Links

LINKS

1. Word Count example - https://medium.com/@gulcanogundur/pyspark-word-count-b099106135a7
2. Spark and Python for Big Data with PySpark - https://www.udemy.com/course/spark-and-python-for-big-data-with-pyspark/
3. Распределенное выполнение Python-задач с использованием Apache Mesos. Опыт Яндекса https://habr.com/ru/company/yandex/blog/306548/