Large-Scale and Multi-Structured Databases

The Big Data Era

Prof Pietro Ducange

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

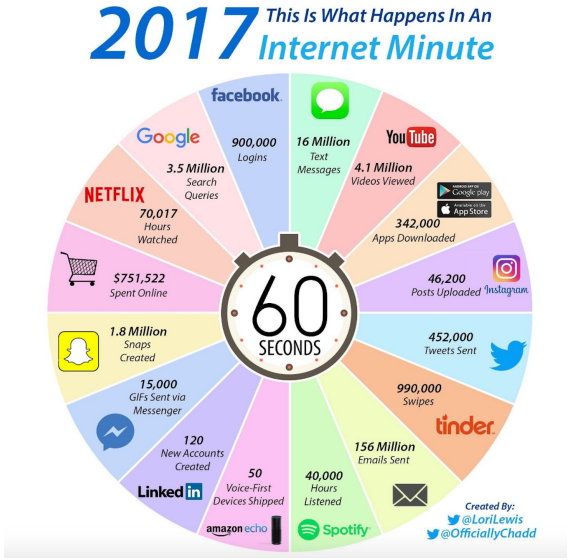
UNIVERSITÀ DI PISA

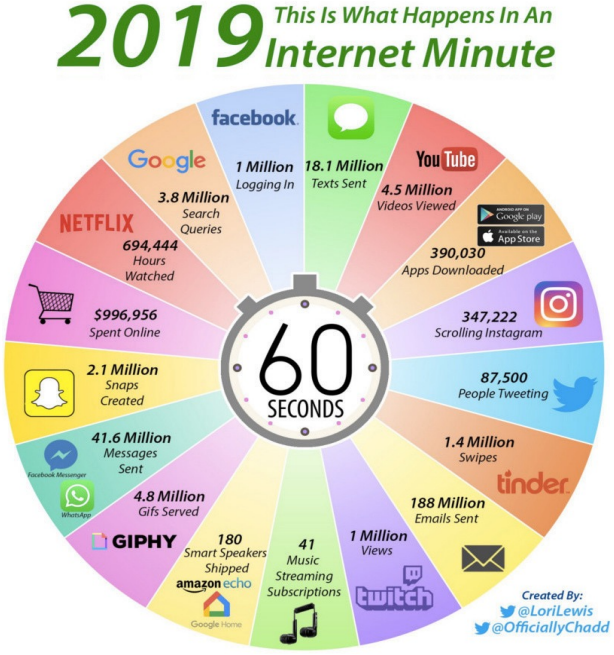
CROSSLAB
Innovation for industry 4.0

Big Data: What, Who, Where, How

- ***Big Data*** is a fashion-trend ***buzz word***
- ***Everybody*** (and ***everywhere***) talk about Big Data
- Enterprises, Universities, Research Centers, Researchers, Software Developers, Consultants, etc., state that they work with Big Data
- ***The question is***: what can we do with Big Data? How can we use Big Data?

The Big Data Era (I)





DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

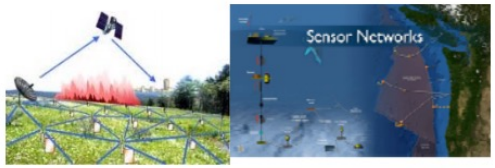
The Big Data Era (II)

Social media and networks
(all of us are generating data)

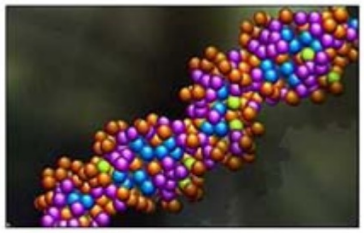
Scientific instruments
(collecting all sorts of data)

Mobile devices
(tracking all objects
all the time)

Transactions



Sensor technology and networks
(measuring all kinds of data)



Genomics

The Big Data Era (II)

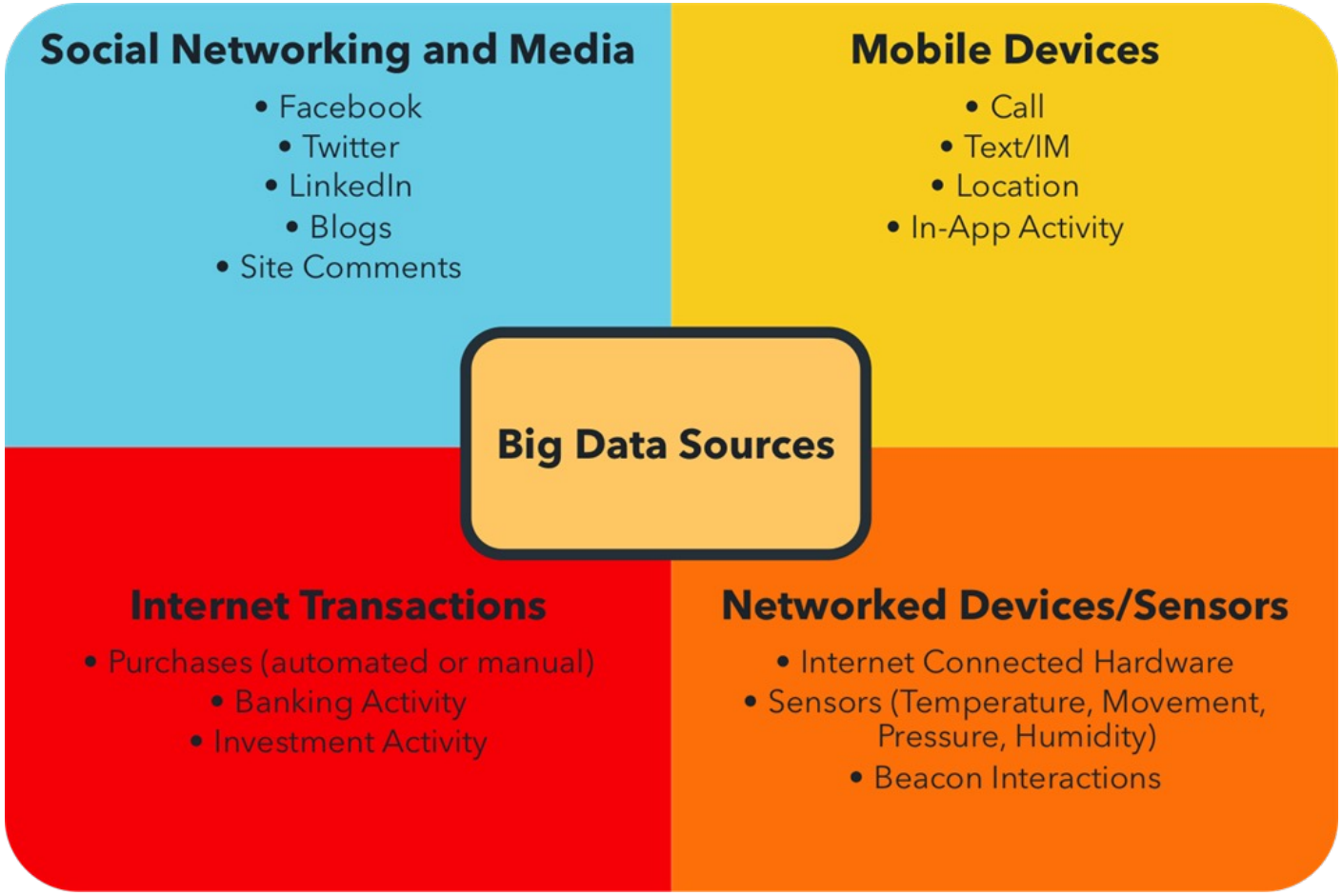


Image extracted from: http://www.talismaniandigital.com/importance-of-big-data-analytics-in-digital-marketing/

Big Data in a Snapshot

Image extracted from: https://www.edureka.co/blog/what-is-big-data/

10 Vs of Big Data

Image extracted from: https://towardsdatascience.com/big-data-analysis-spark-and-hadoop-a11ba591c057

42 Vs of Big Data (in 2017)

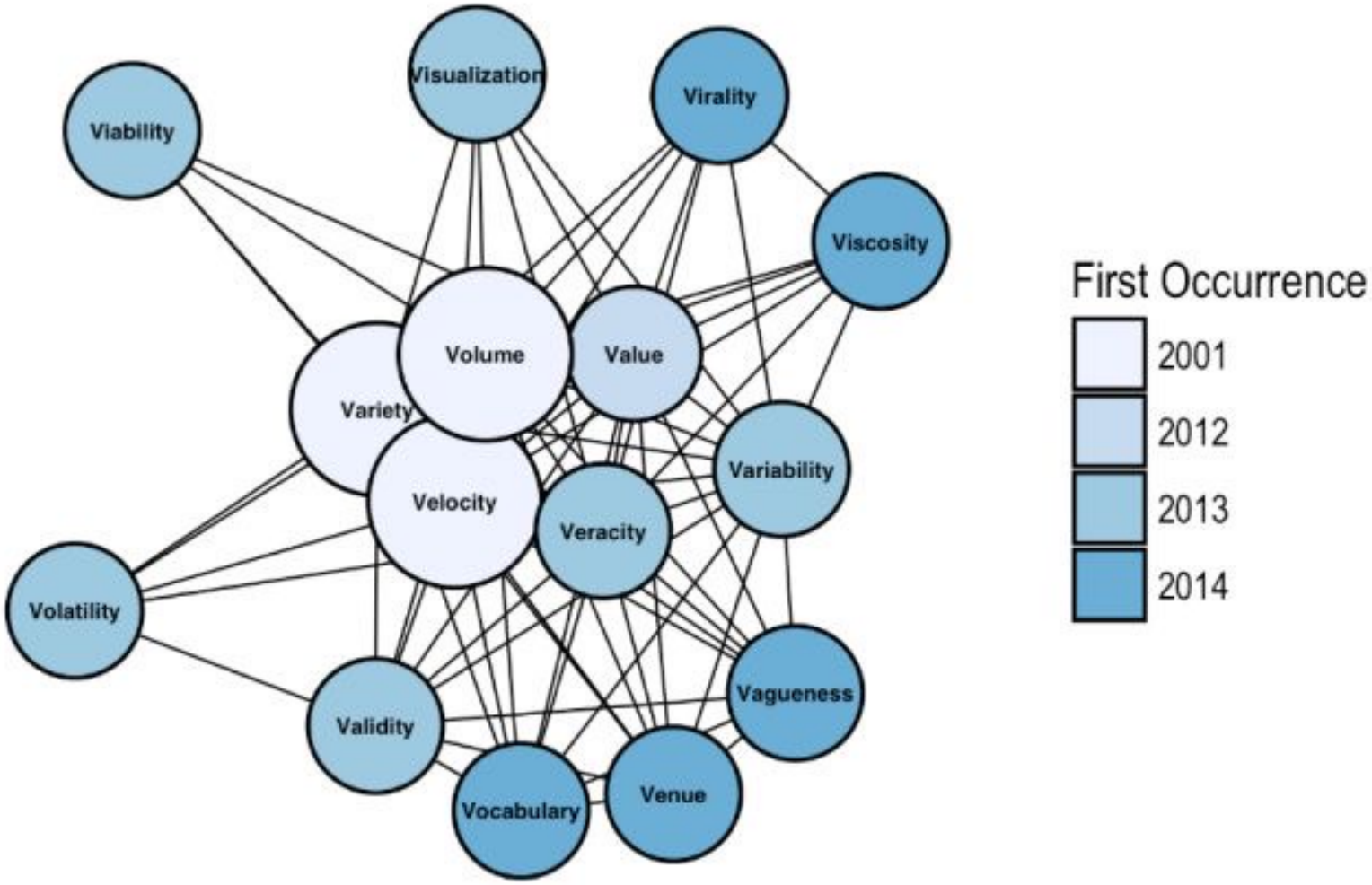
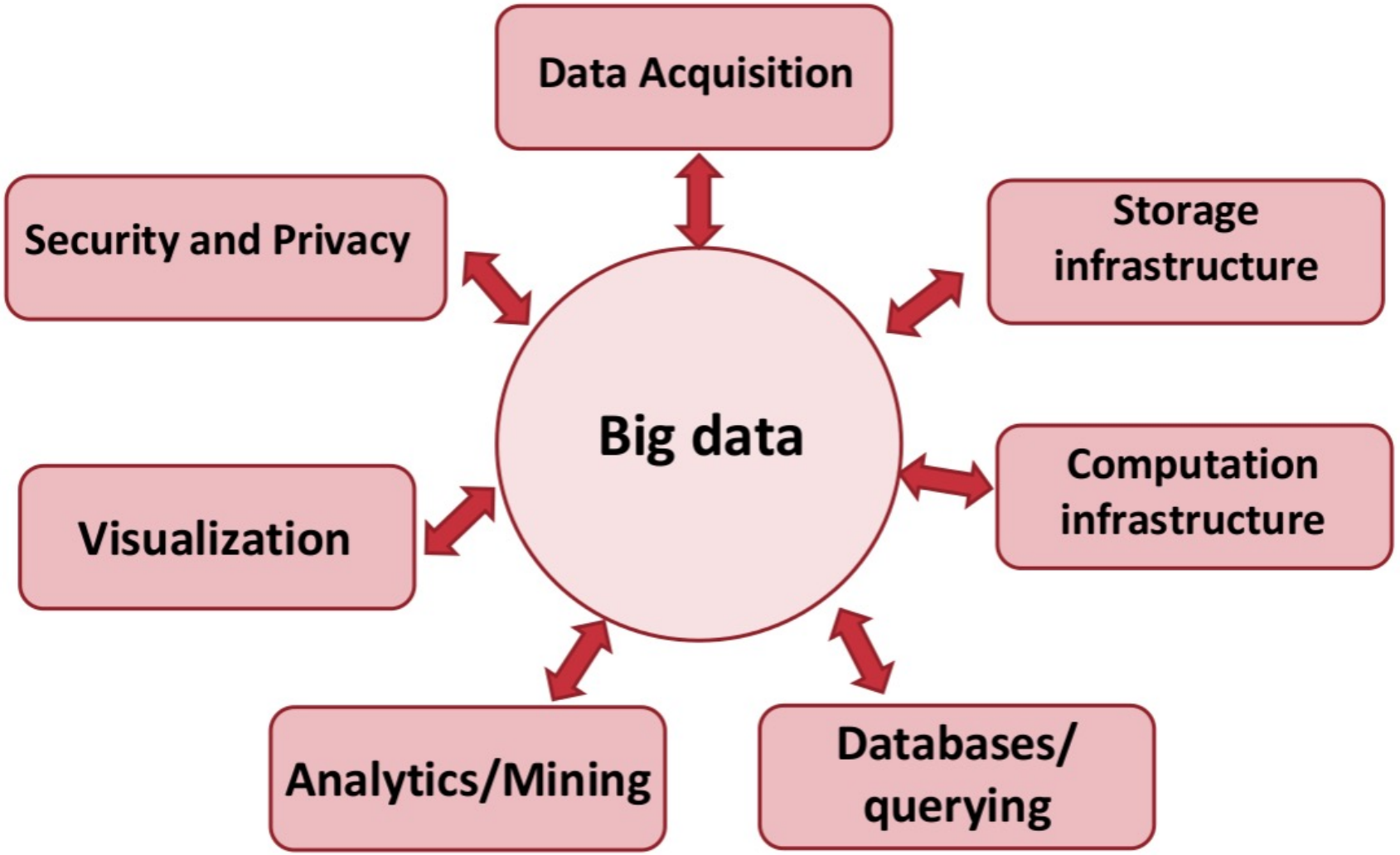


Image extracted from: https://www.kdnuggets.com/2017/04/42-vs-big-data-data-science.html

Big data has many faces



Source: Yvan Saeys. A gentle Introduction to Big Data

The Main Big Data Issue

Big data refers to any problem characteristic that represents a ***challenge*** to process it with ***traditional applications***

Big Data involves ***data*** whose volume, diversity and complexity requires ***new techniques, algorithms and analyses*** to fetch, to store and to elaborate them.

The Main Big Data Computing Issue

- ***Classical*** data elaboration algorithms, such as the ones for analytics and data mining, cannot directly applied to Big Data

- **New** technologies are needed for both ***storage*** and ***computational*** tasks.

- ***New paradigms*** are needed for ***designing***, ***implementing*** and ***experimenting*** algorithms for handling big data.

How to deal with data intensive applications?

Scale-up vs. Scale-out

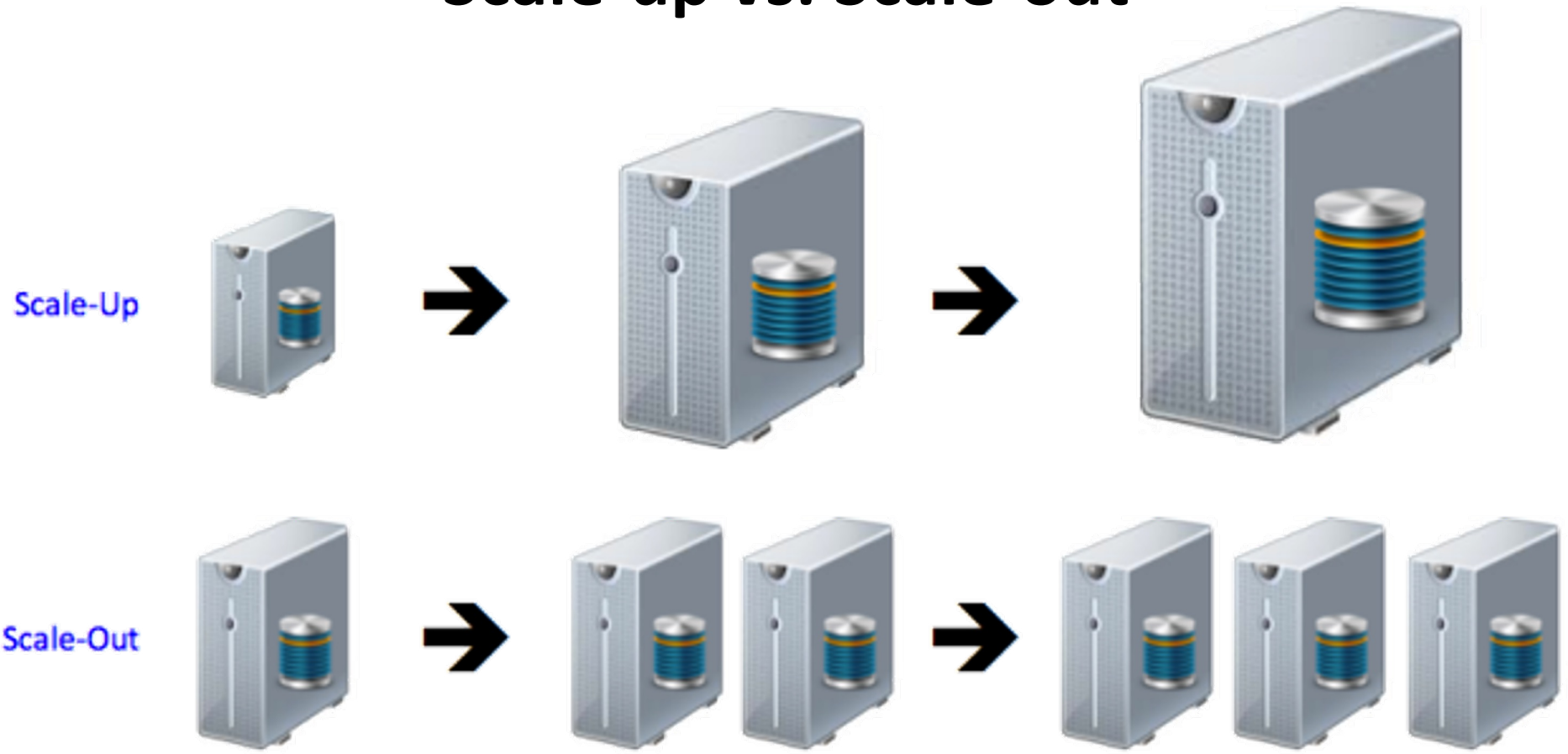
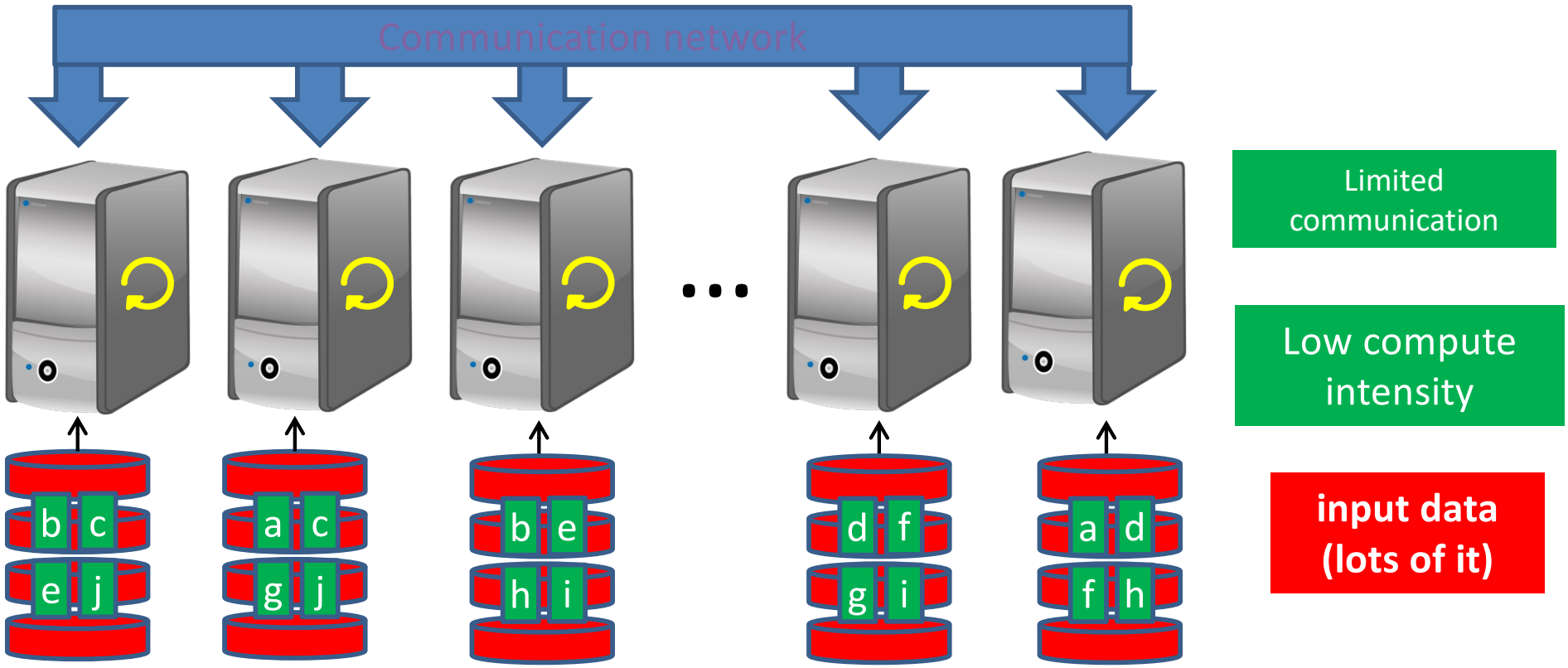


Image extracted from: https://hadoop4usa.wordpress.com/2012/04/13/scale-out-up/

Distributed systems in Big Data

- **Objective: To apply an operation to all data**
  - One machine cannot process or store all data
    - Data is ***distributed*** in a cluster of computing nodes
    - It does ***not matter*** which machine executes the operation
    - It does not matter if it ***is run twice*** in different nodes (due to failures or straggler nodes)
    - We look for an ***abstraction of the complexity*** behind distributed systems
  - **DATA LOCALITY *is crucial***
    - Avoid data transfers between machines as much as possible

Slide courtesy of F. Herrera, A. Fernandez, Isaac Triguero, Fuzzy Models for Data Science and Big Data, Tutorial @ FuzzIEEE 2017, Naples, Italy



Solution: store data on local disks of the nodes that perform computations on that data ("**data locality**")

Slide courtesy of F. Herrera, A. Fernandez, Isaac Triguero, Fuzzy Models for Data Science and Big Data, Tutorial @ FuzzIEEE 2017, Naples, Italy

MapReduce (I)

New programming model: *MapReduce*

- *"Moving computation is cheaper than moving computation and data at the same time"*
- ***Idea***
  - ***Data*** is ***distributed*** among nodes (distributed file system)
  - ***Functions/operations*** to process data are ***distributed*** to all the computing nodes
  - Each computing node works with the data stored in it
  - ***Only the necessary data is moved across the network***

Slide courtesy of F. Herrera, A. Fernandez, Isaac Triguero, Fuzzy Models for Data Science and Big Data, Tutorial @ FuzzIEEE 2017, Naples, Italy

MapReduce (II)

- Parallel Programming model
- Introduce by Google in 2004
- **Divide & conquer strategy**
  - **divide**: partition dataset into smaller, independent chunks to be processed in parallel (*map*)
  - **conquer**: combine, merge or otherwise aggregate the results from the previous step (*reduce*)

MapReduce: Basic Working (I)

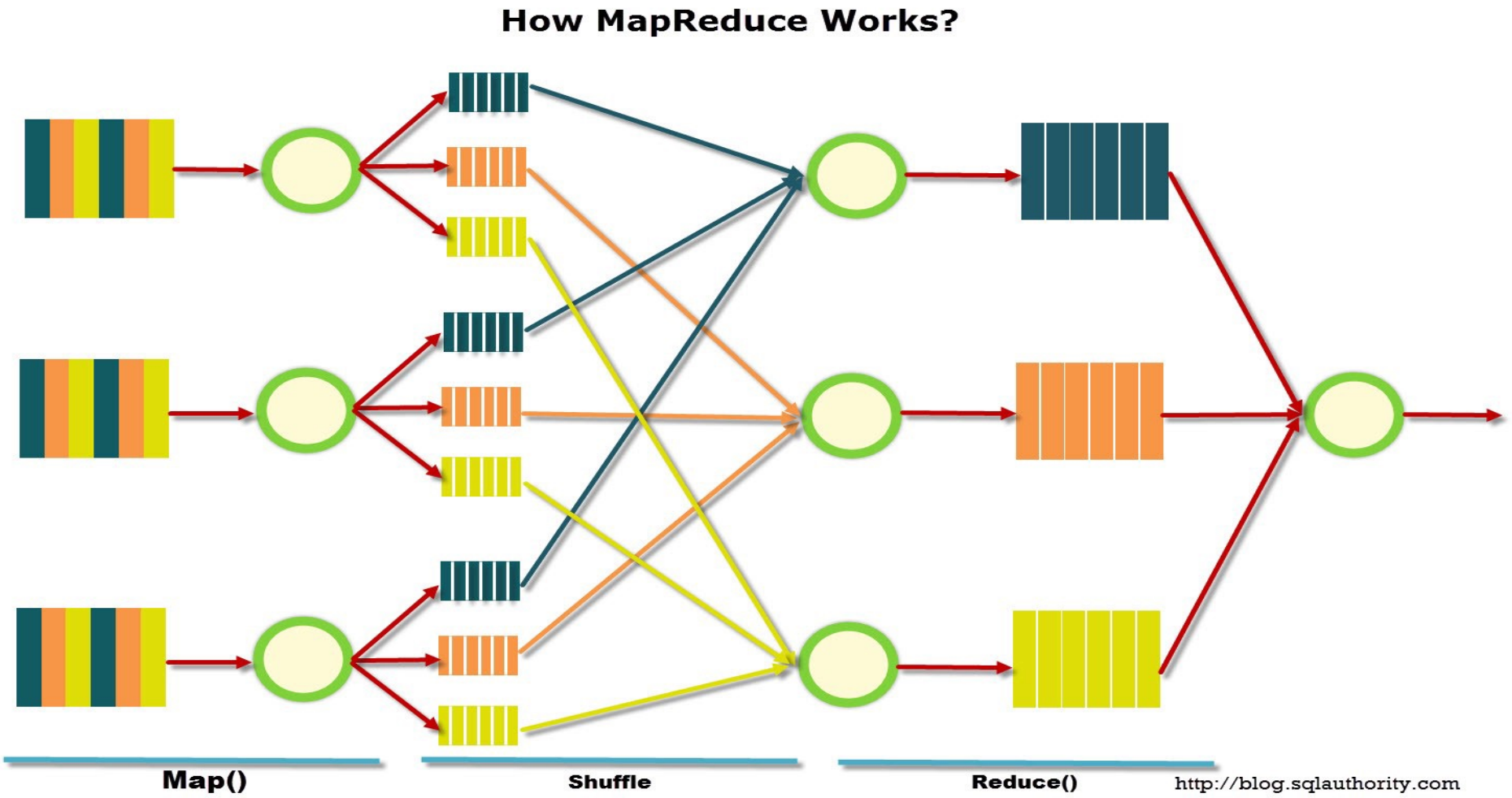


Image extracted from: Elkano, Mikel, et al. "CHI-BD: A fuzzy rule-based classification system for Big Data classification problems." Fuzzy Sets and Systems (2017).

MapReduce: Basic Working (II)

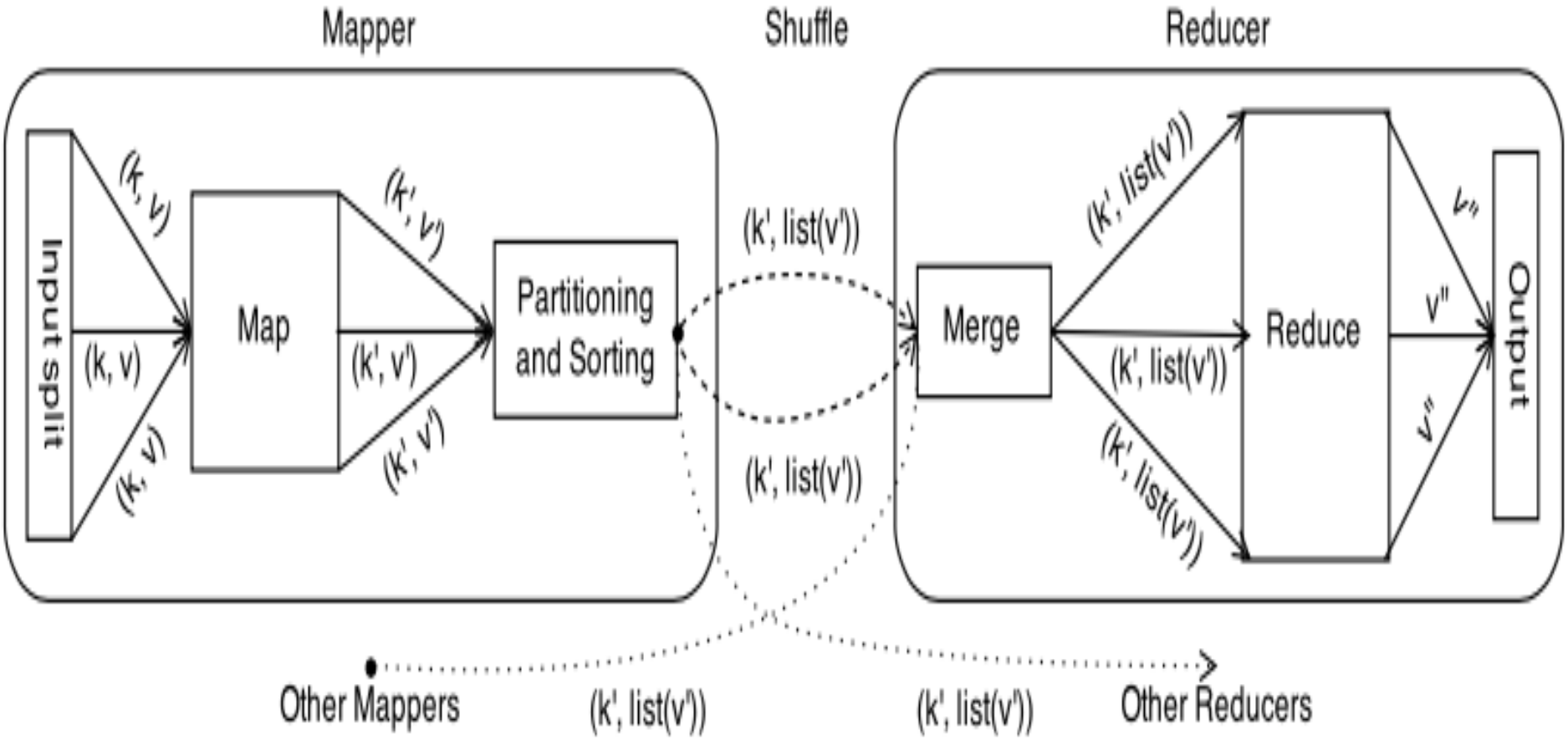
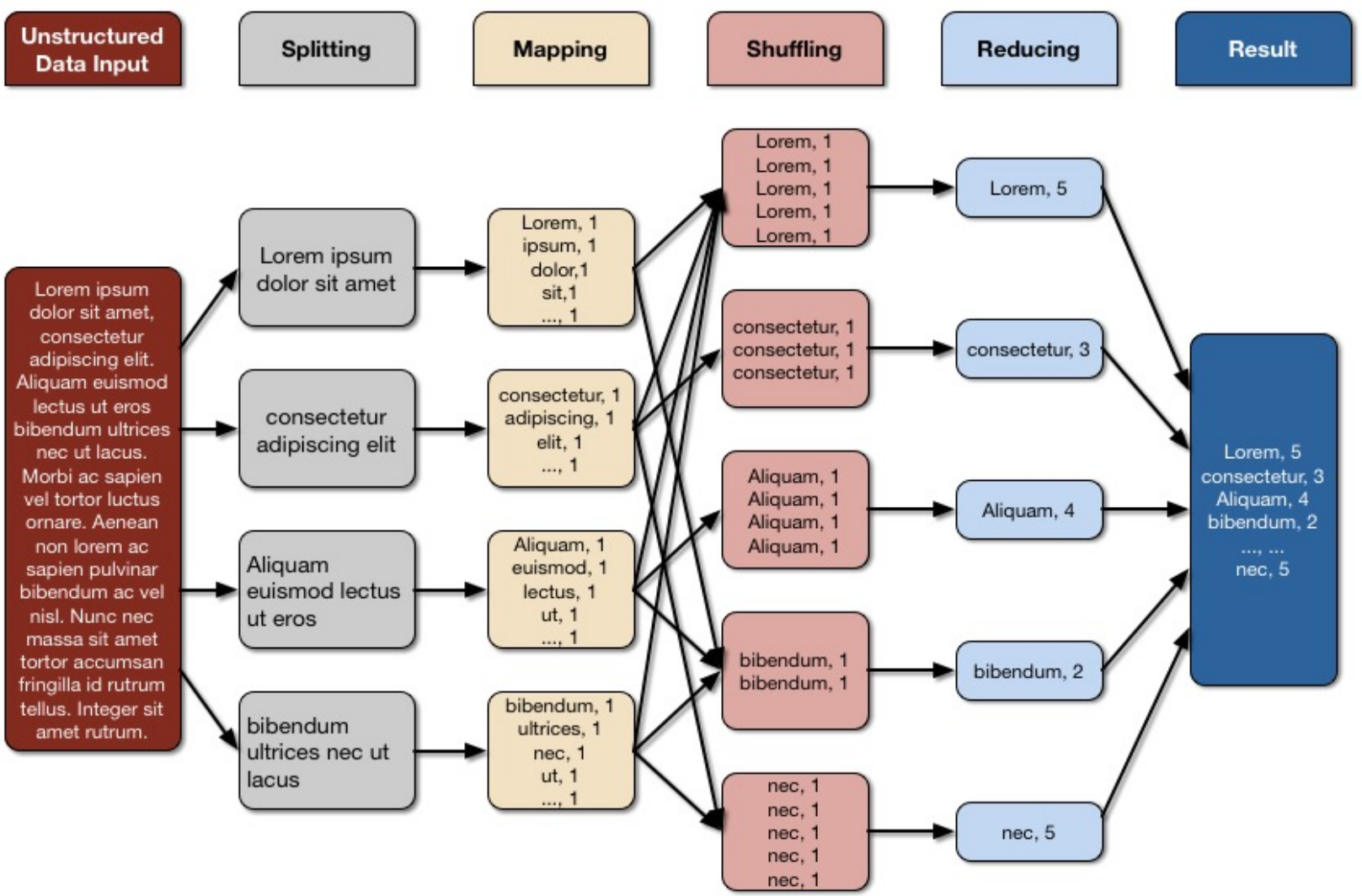


Image extracted from: Elkano, Mikel, et al. "CHI-BD: A fuzzy rule-based classification system for Big Data classification problems." Fuzzy Sets and Systems (2017).

MapReduce: WordCount Example



Google Software Architecture for Big Data

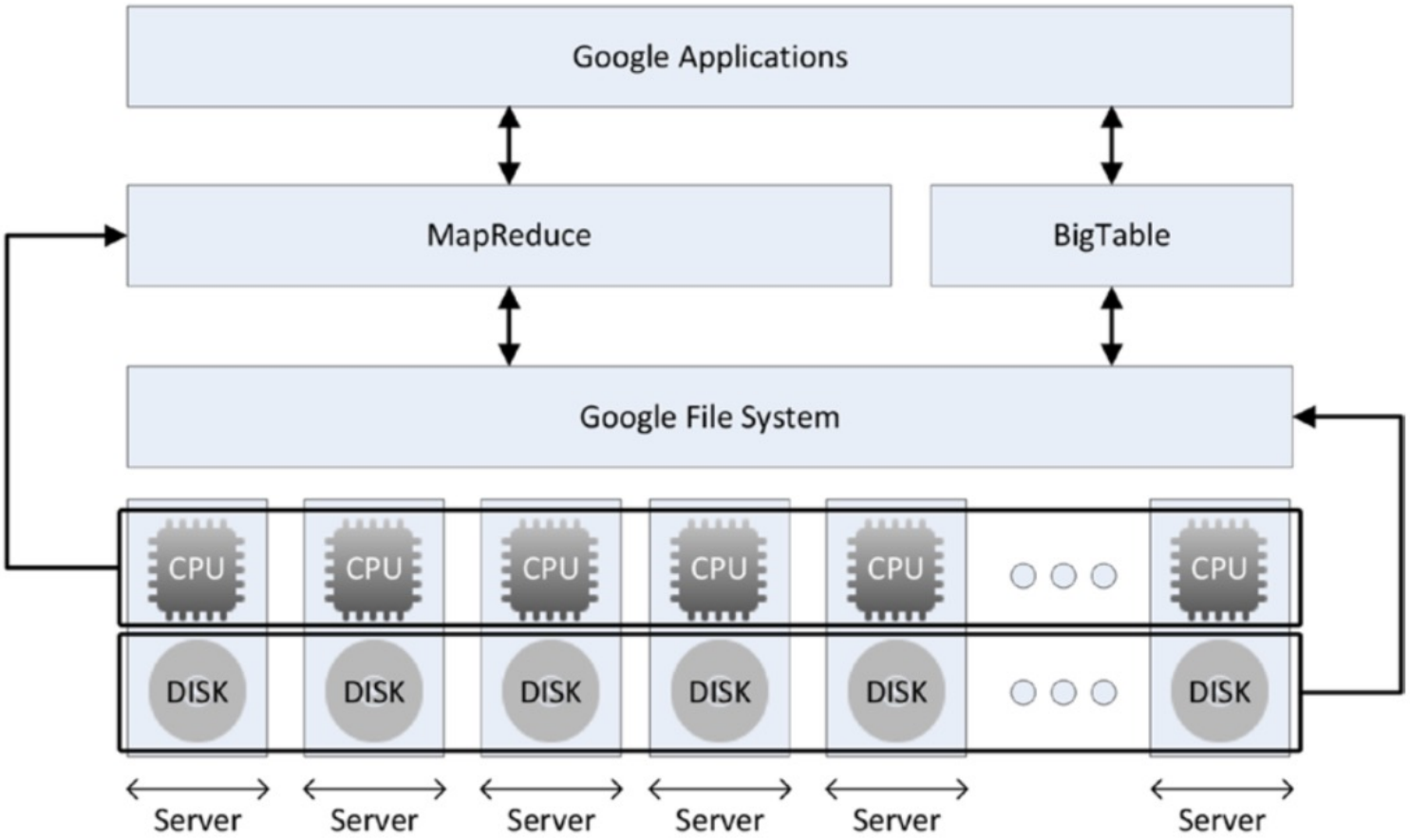


Image extracted from: "*Guy Harrison, Next Generation Databases, Apress, 2015*"

Hadoop

- Hadoop is:
  - An **open-source** (the first) framework written in Java
  - Distributed storage of very large data sets (Big Data)
  - Distributed processing of very large data sets

- This framework consists of a **number of modules**
  - ***Hadoop Common****: the common utilities that support the other Hadoop modules.*
  - ***Hadoop Distributed File System (HDFS)***
  - **Hadoop YARN** – A framework for job scheduling and cluster resource management
  - ***Hadoop MapReduce*** – programming model

http://hadoop.apache.org/

Distributed File System: HDFS

HDFS – Hadoop Distributed File System

- Distributed File System written in Java
- Scales to clusters with **thousands of computing nodes**
  - Each node stores part of the data in the system
- **Fault tolerant** due to data replication
- Designed for big files and low-cost hardware
  - GBs, TBs, PBs
- **Efficient for read and append operations** (random updates are rare)

Hadoop Limits

- Hadoop is optimized for ***one-pass batch*** processing of on-disk data

- It suffers for ***interactive data exploration*** and more complex multi-pass analytics algorithms

- Due to a ***poor inter-communication*** capability and inadequacy for ***in-memory computation*** , Hadoop ***is not suitable*** for those applications that require ***iterative and/or online computation***

SPARK

- ***Apache Spark*** is an open-source which has emerged as the next generation big data processing tool due to its enhanced ***flexibility*** and ***efficiency***.

- Spark allows employing ***different distributed programming models***, such as ***MapReduce*** and ***Pregel***, and has proved to perform ***faster*** than Hadoop , especially in case of ***iterative and online applications***.

- Unlike the disk-based MapReduce paradigm supported by Hadoop, Spark employs the concept of ***in-memory cluster computing***, where datasets are cached in memory to reduce their access latency.

Driver and Executors

At high level, a Spark application runs as a set of ***independent processes*** on the top of the dataset distributed across the machines of the cluster and consists of one driver program and several executors

The ***driver program***, hosted in the master machine, ***runs*** the user's ***main function*** and ***distributes*** operations on the cluster by sending several units of work, called tasks, to the executors.

Each executor, hosted in a ***slave machine***, runs tasks in ***parallel*** and keeps data in memory or disk storage across them.

Driver and Executors

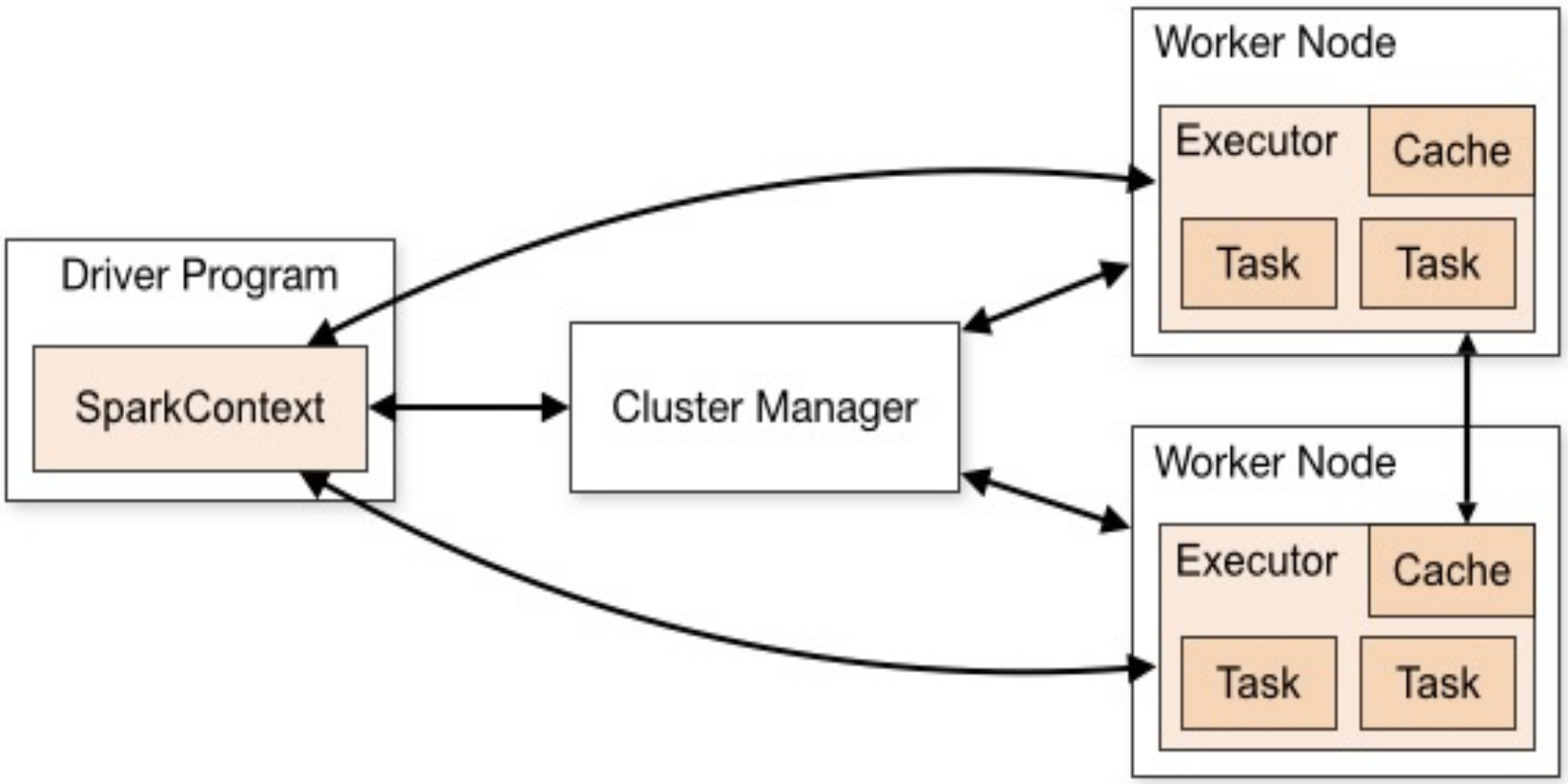


Image extracted from: https://spark.apache.org/docs/latest/cluster-overview.html

SPARK RDD

- The main abstraction provided by Spark is the ***resilient distributed dataset*** (RDD)
- RDD is a ***fault-tolerant*** collection of elements partitioned across the machines of the cluster that can be processed in parallel
- These collections are ***resilient***, because they ***can be rebuilt*** if a portion of the dataset is lost
- The applications developed using the Spark framework are ***totally independent*** of the ***file system*** or the ***database management*** system used for storing data
- Indeed, there exist ***connectors*** for reading data, creating the RDD and writing back results on files or on databases
- In the last years, ***Data Frames*** and ***Datasets*** have been recently released as an abstraction on top of the RDD.

Spark Ecosystem

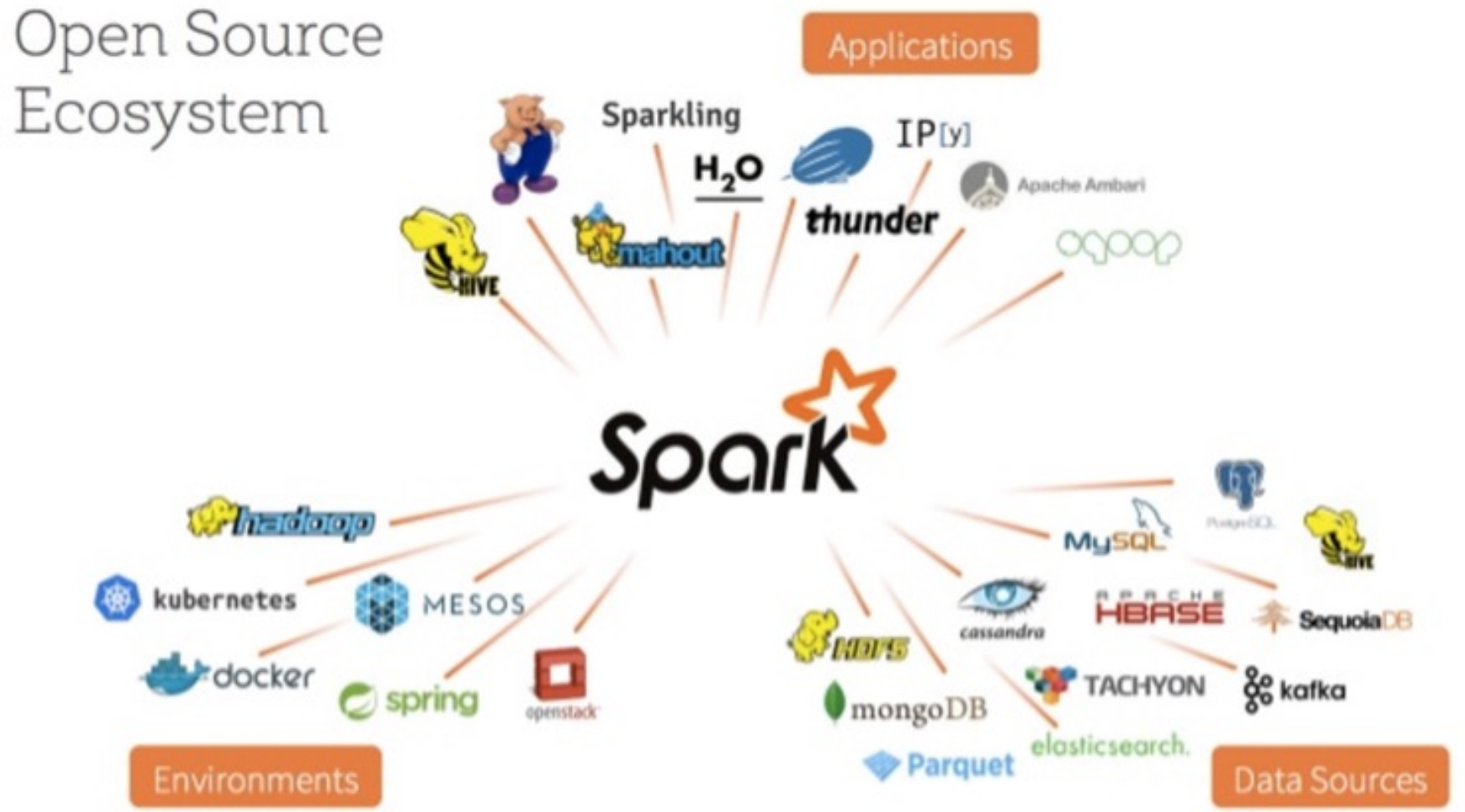
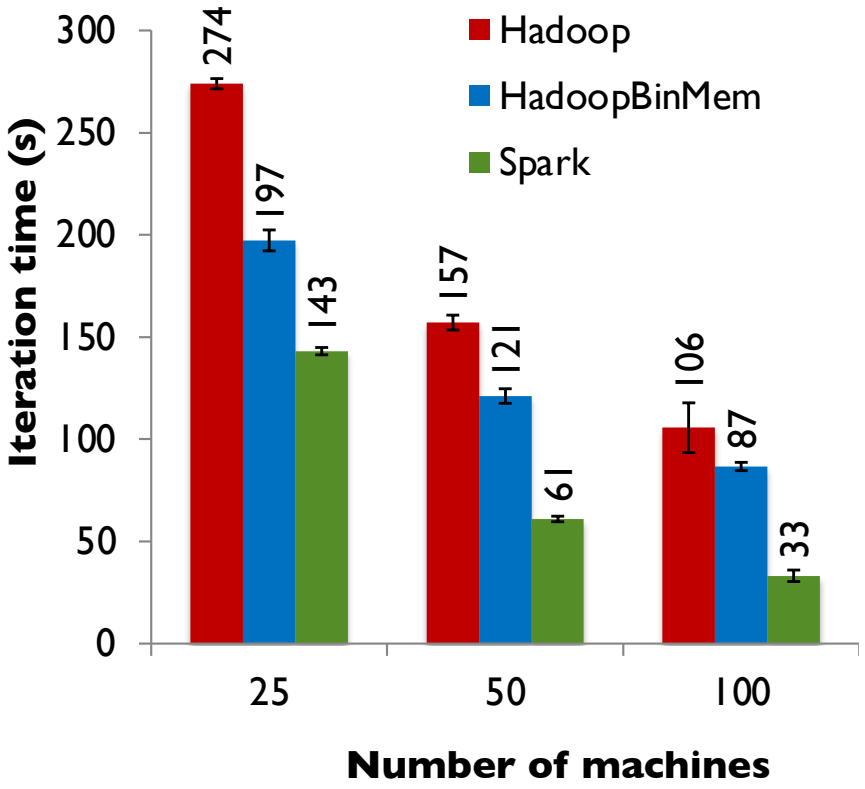


Image extracted from: https://www.zdnet.com/article/the-future-of-the-future-spark-big-data-insights-streaming-and-deep-learning-in-the-cloud/

Hadoop vs SPARK

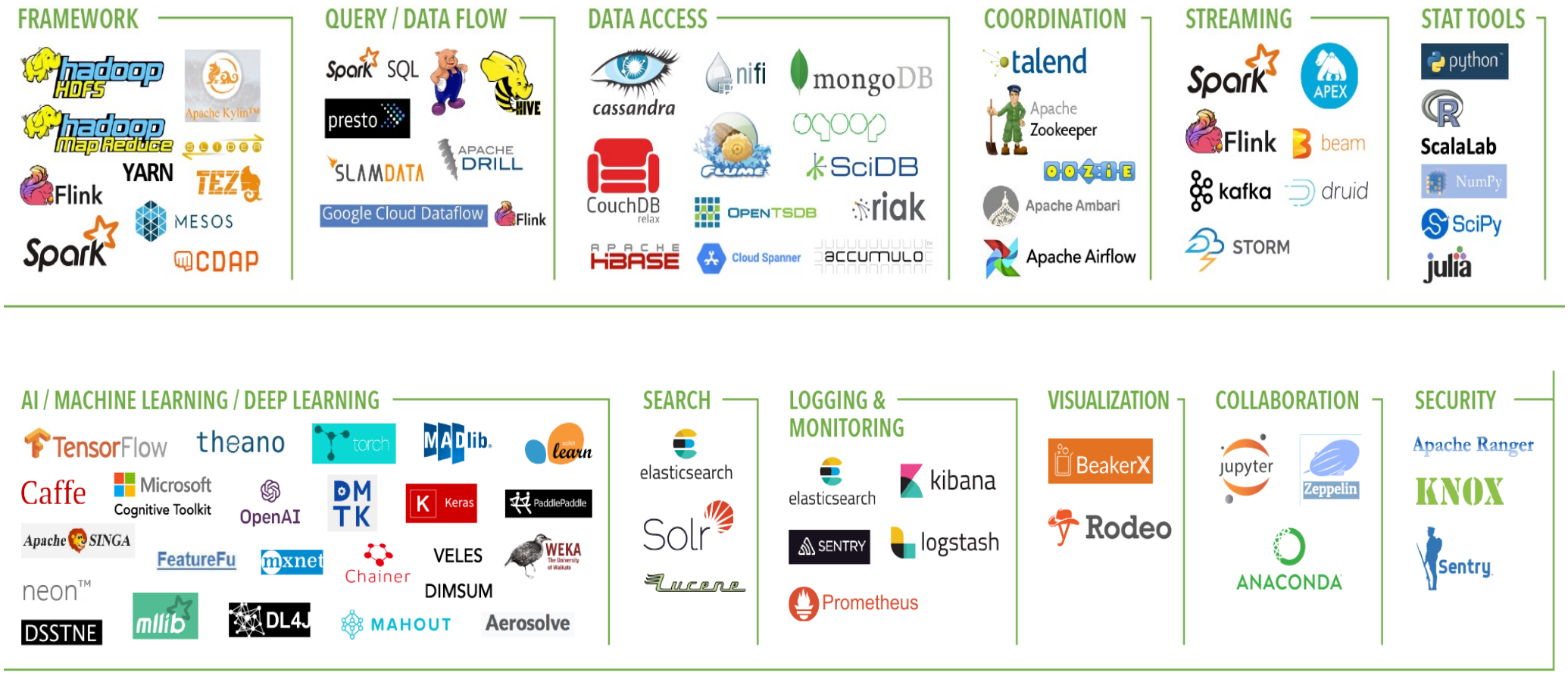
Up to **10×** faster on disk,
100× in memory

2-5× less code



More details can be found in: Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.

Open source solutions for Big Data

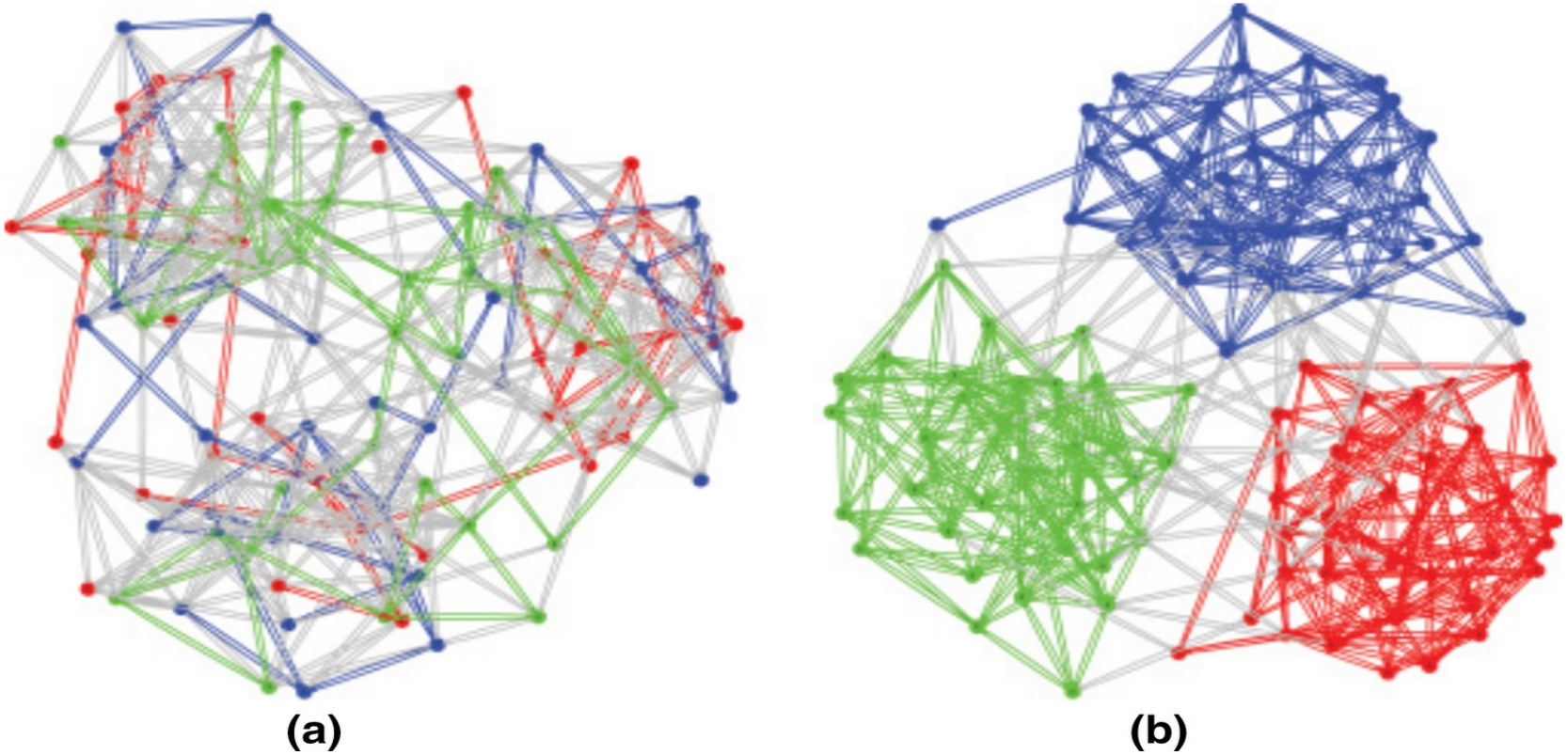


DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB
Innovation for industry 4.0

A distributed algorithm for large-scale graph partitioning

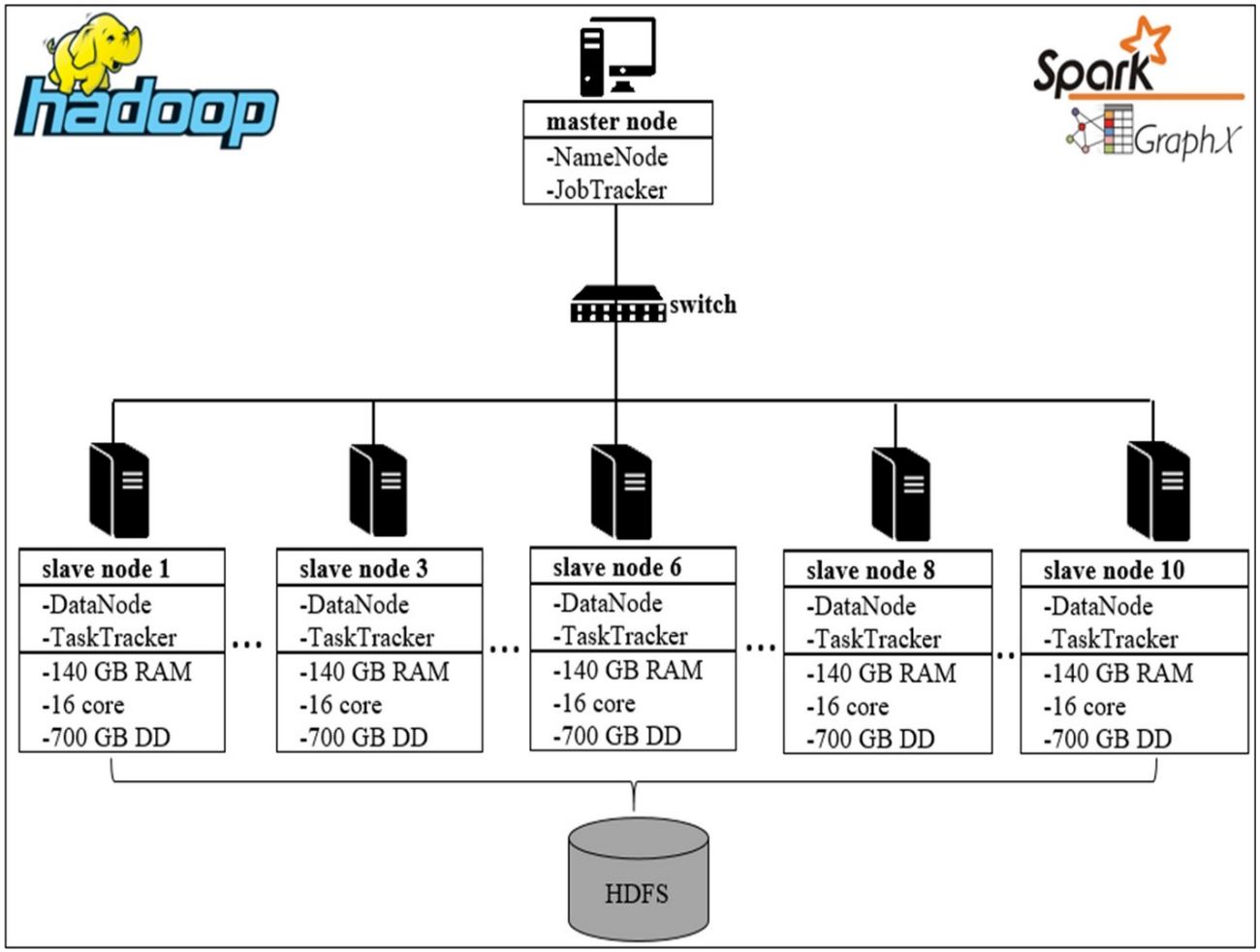


Shoddy Partitioning

Optimal Partitioning

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00357-y

A distributed algorithm for large-scale graph partitioning



Designation	Type	|*Vr*|	|*Ed*|
Biosnap	Undirected	1018524	24735503
Twitter	Directed	81306	1768149
Usroad	Directed	126146	161950
Email	Undirected	36692	183831
Astro	Undirected	18772	198110
ageRRN	Directed	860000	2360000
cptTRM	Directed	1070000	6000000
elsaRR	Directed	1508000	9000000
osmMA	Directed	4526700	12670000

Big Data architecture for intelligent maintenance

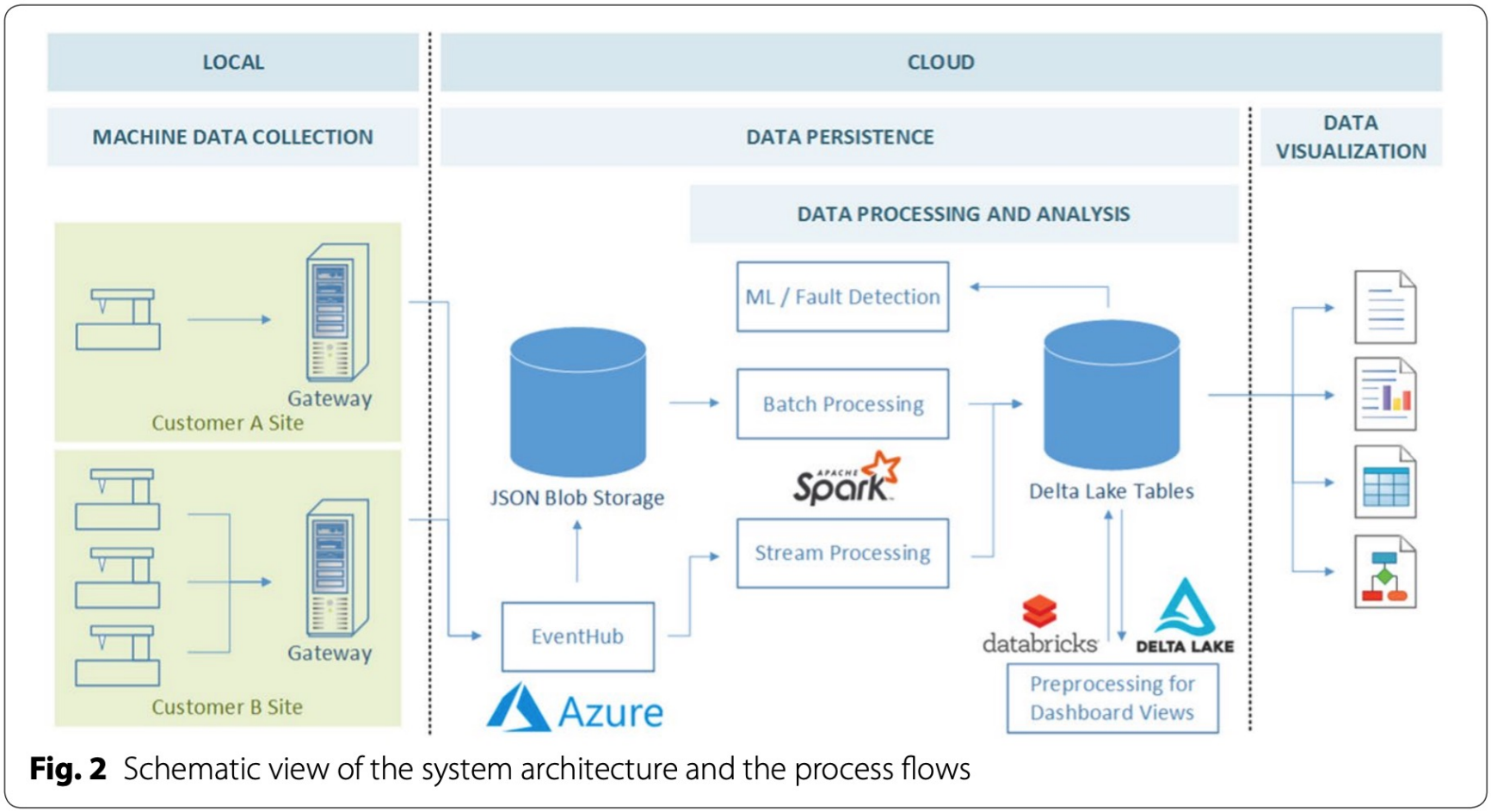


Fig. 2 Schematic view of the system architecture and the process flows

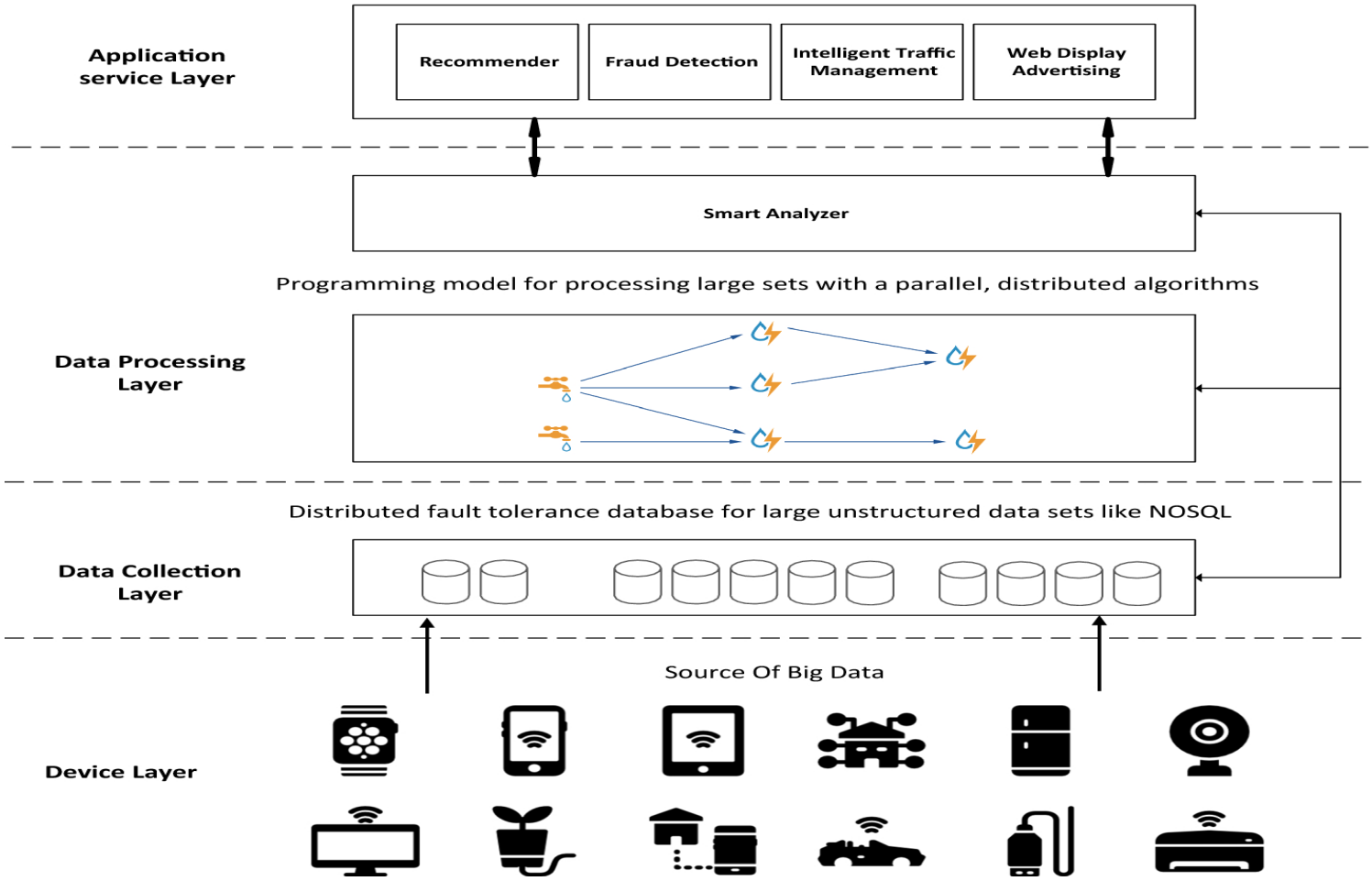
https://link.springer.com/article/10.1186/s40537-020-00340-7

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities



https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0215-2

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

Suggested Readings

Chapter 2 of the book "*Guy Harrison, Next Generation Databases, Apress, 2015*"

https://hadoop.apache.org/

https://spark.apache.org/