Introducción a big data y big data analytics

Francisco Herrera

Research Group on Soft Computing and Information Intelligent Systems (SCI²S)

http://sci2s.ugr.es

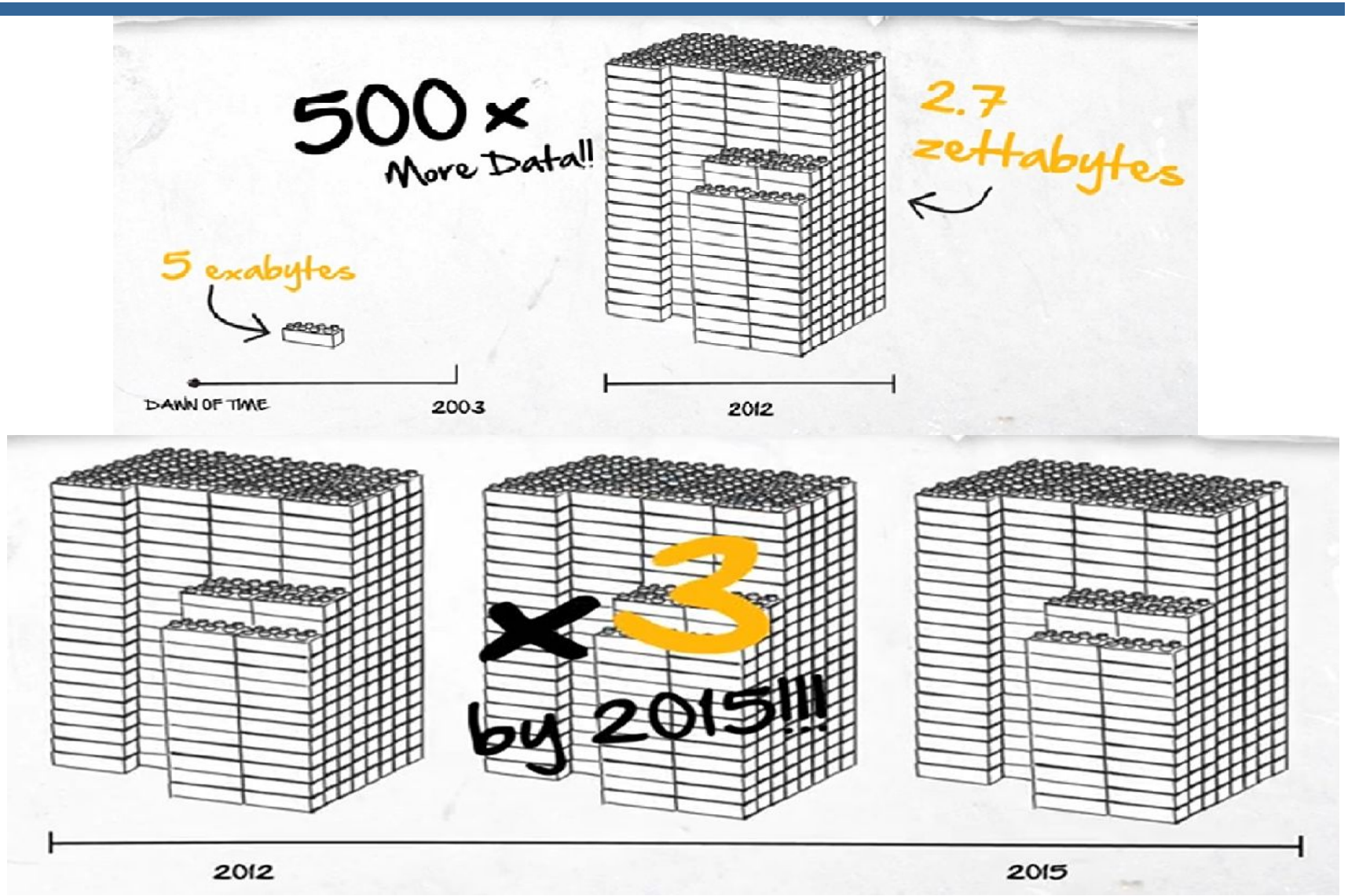
Dept. of Computer Science and A.I.

University of Granada, Spain

Email: herrera@decsai.ugr.es

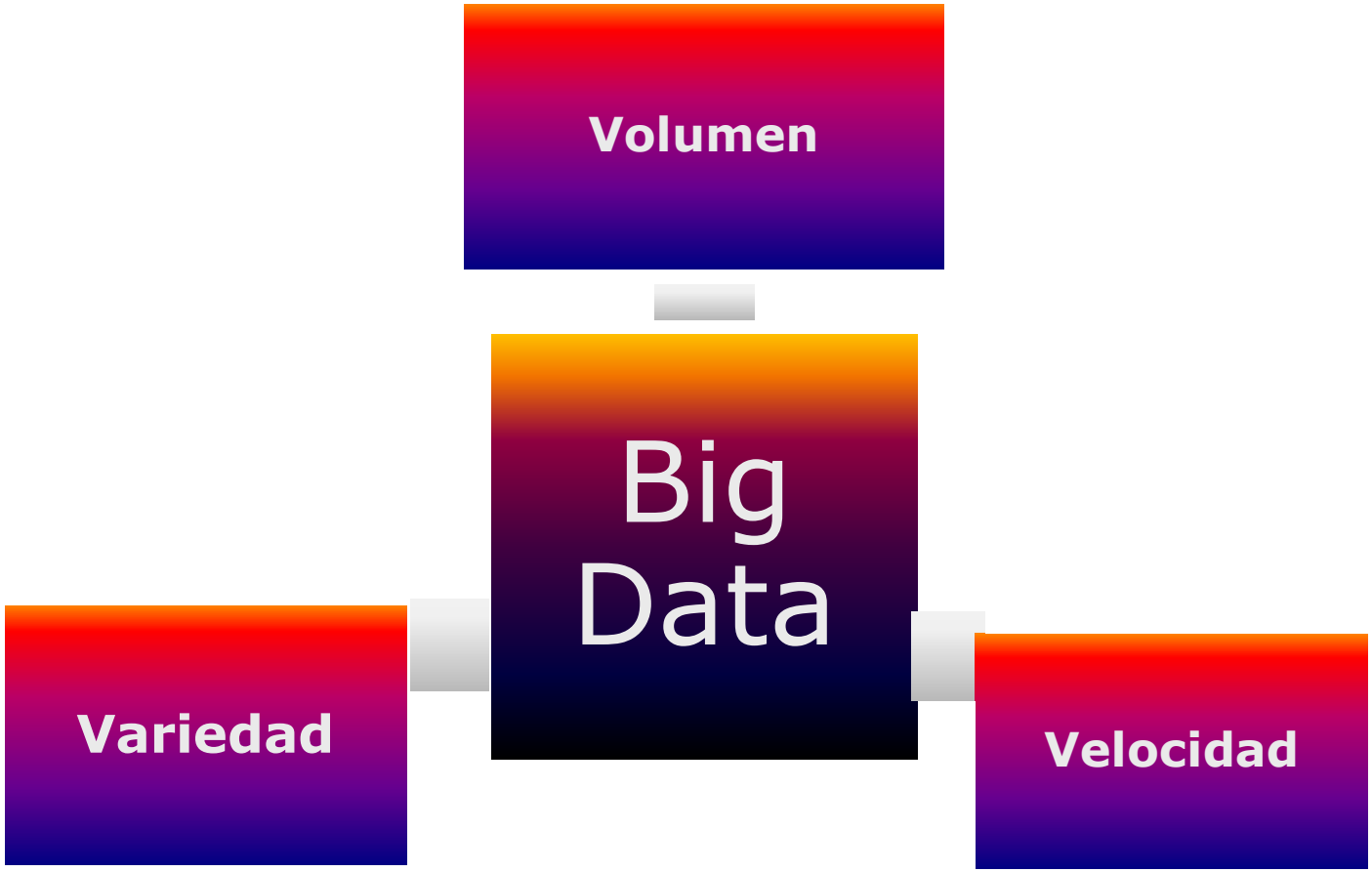
DECSAI
Universidad de Granada

Big Data: La explosión de los datos



Big Data

Big Data en 3 V's



Doug Laney, Gartner 2001

Índice

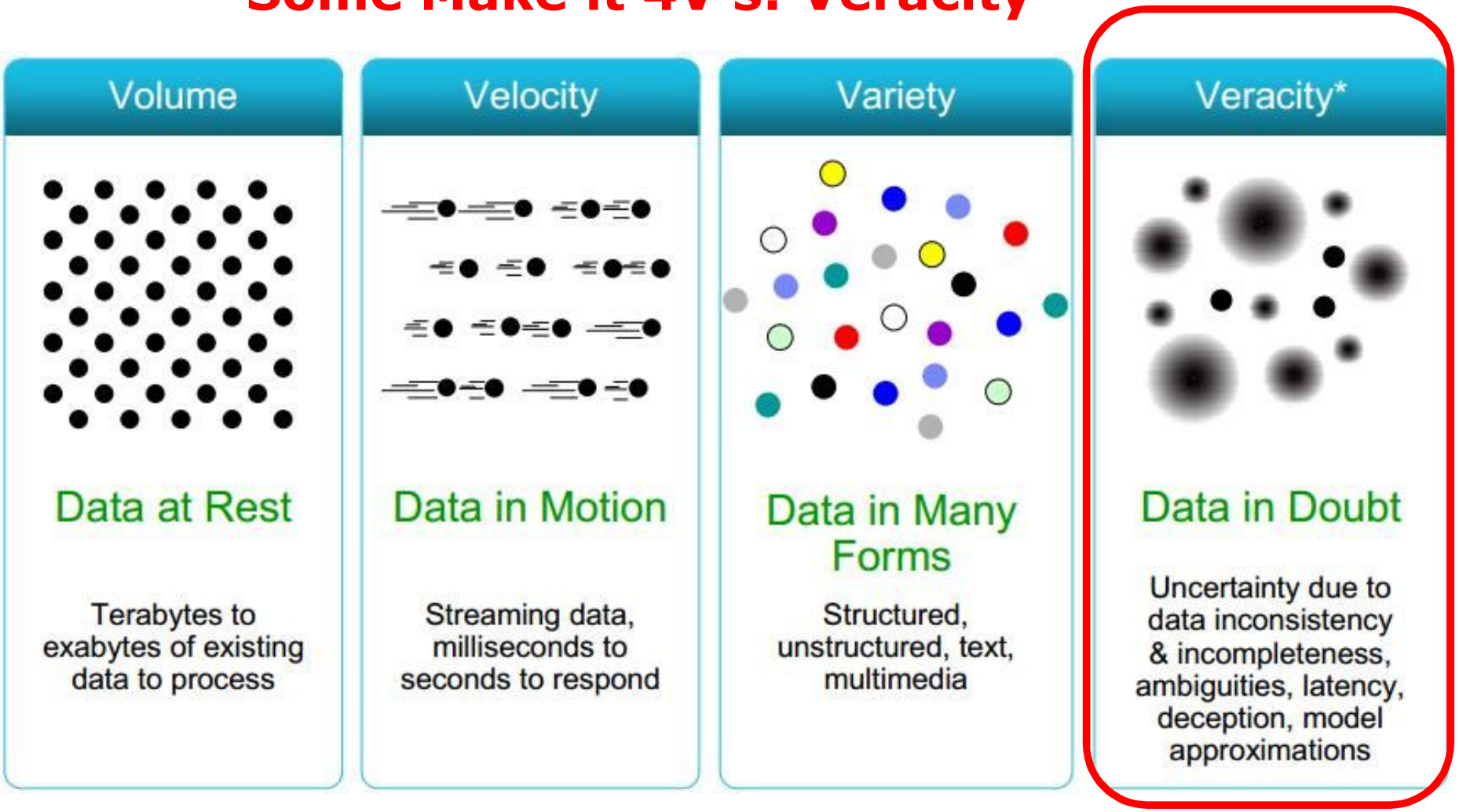
- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

Índice

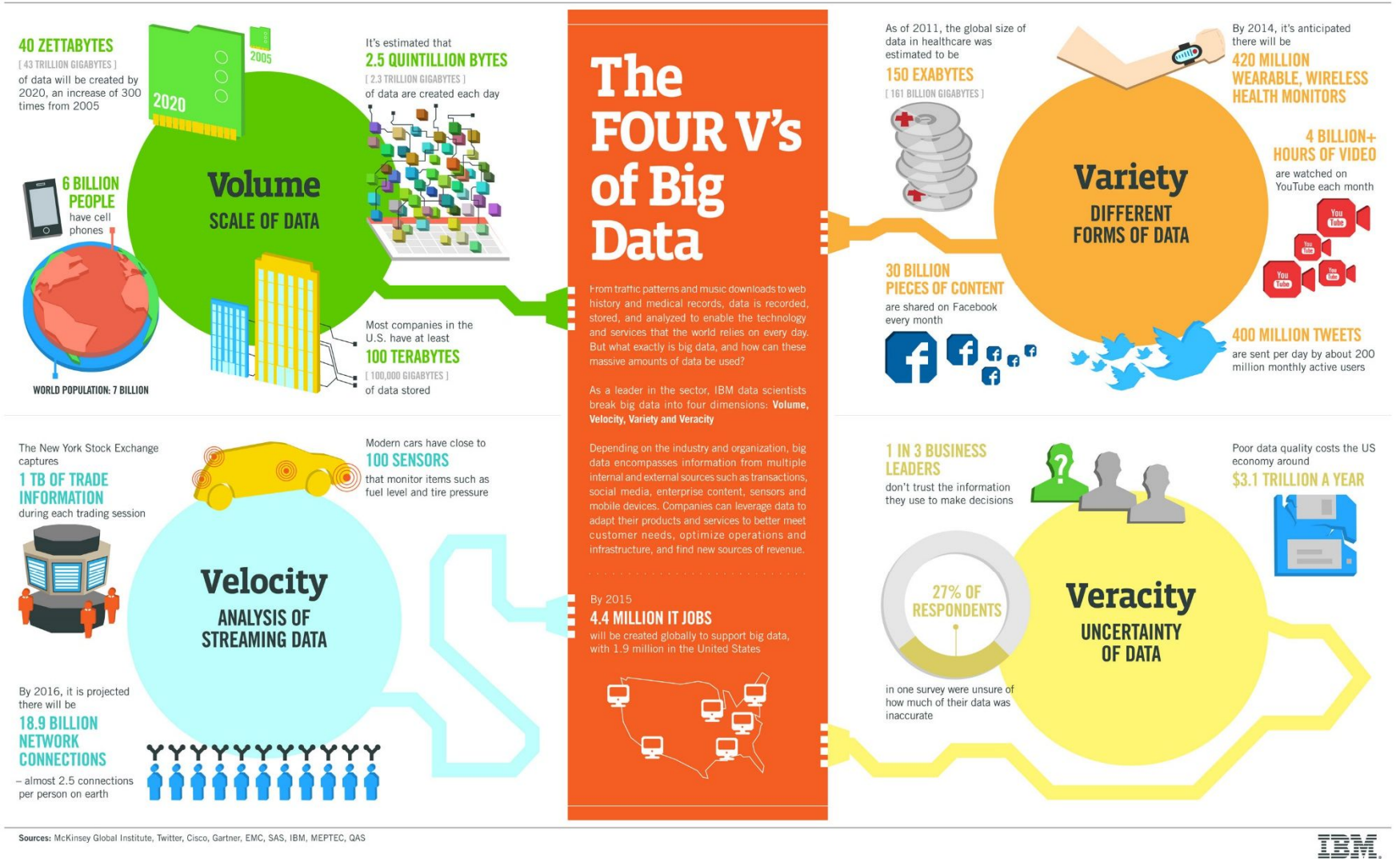
- **Big Data. Big Data Science**
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

¿Qué es Big Data? 3 V's de Big Data

Some Make it 4V's: Veracity

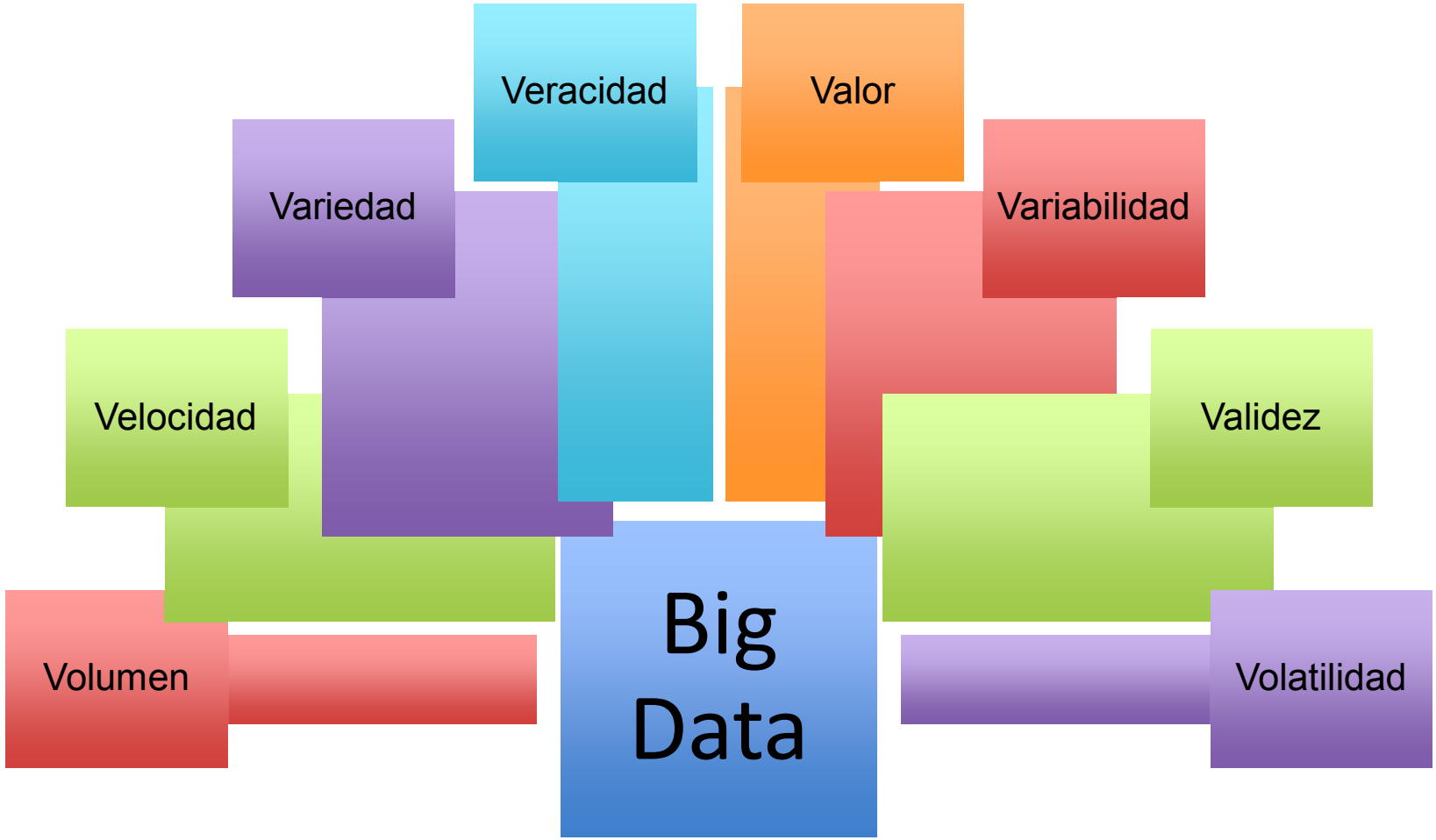


¿Qué es Big Data?



¿Qué es Big Data?

Las 8 V's de Big Data

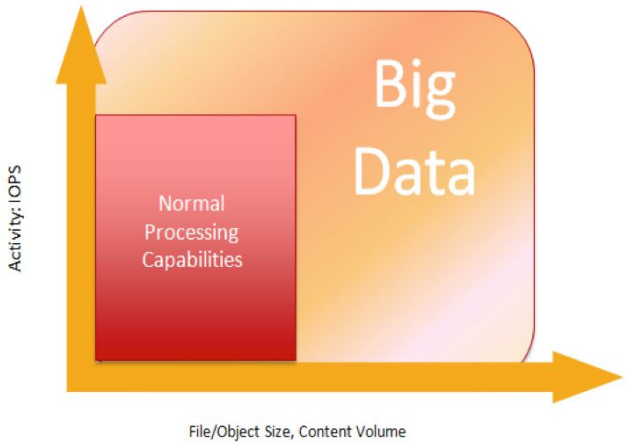


¿Qué es Big Data?

No hay una definición estándar

Big data es una colección de datos grande, complejos, **muy difícil de procesar a través de herramientas de gestión y procesamiento de datos tradicionales**

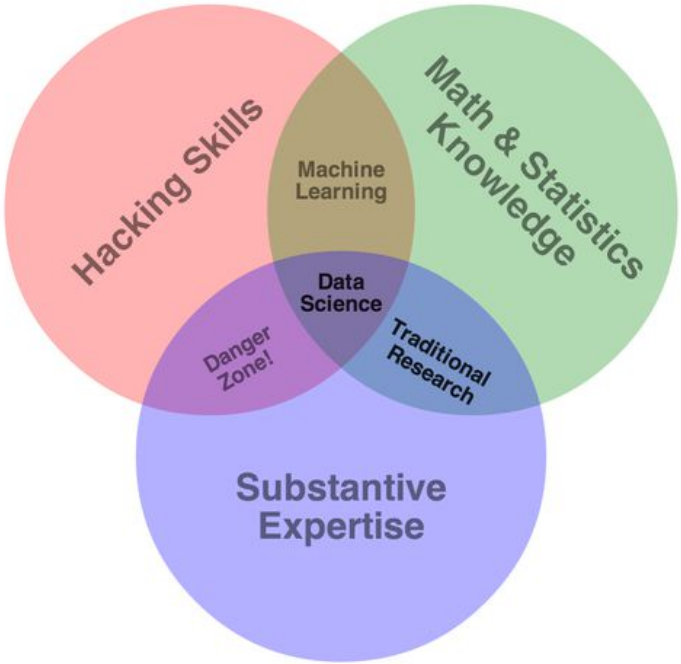
"***Big Data***" son datos cuyo volumen, diversidad y complejidad **requieren nueva arquitectura, técnicas, algoritmos y análisis** para gestionar y extraer valor y conocimiento oculto en ellos ...



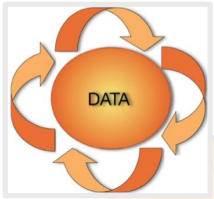
(Big) Data Science

Data Science combines the traditional scientific method with the ability to explore, learn and gain deep insight for (Big) Data

It is not just about finding patterns in data … it is mainly about explaining those patterns



Data Science Process



Data Preprocessing

- Clean
- Sample
- Aggregate
- Imperfect data: missing, noise, …
- Reduce dim.
- …

> 70% time!

Data Processing

- Explore data
- Represent data
- Link data
- Learn from data
- Deliver insight
- …

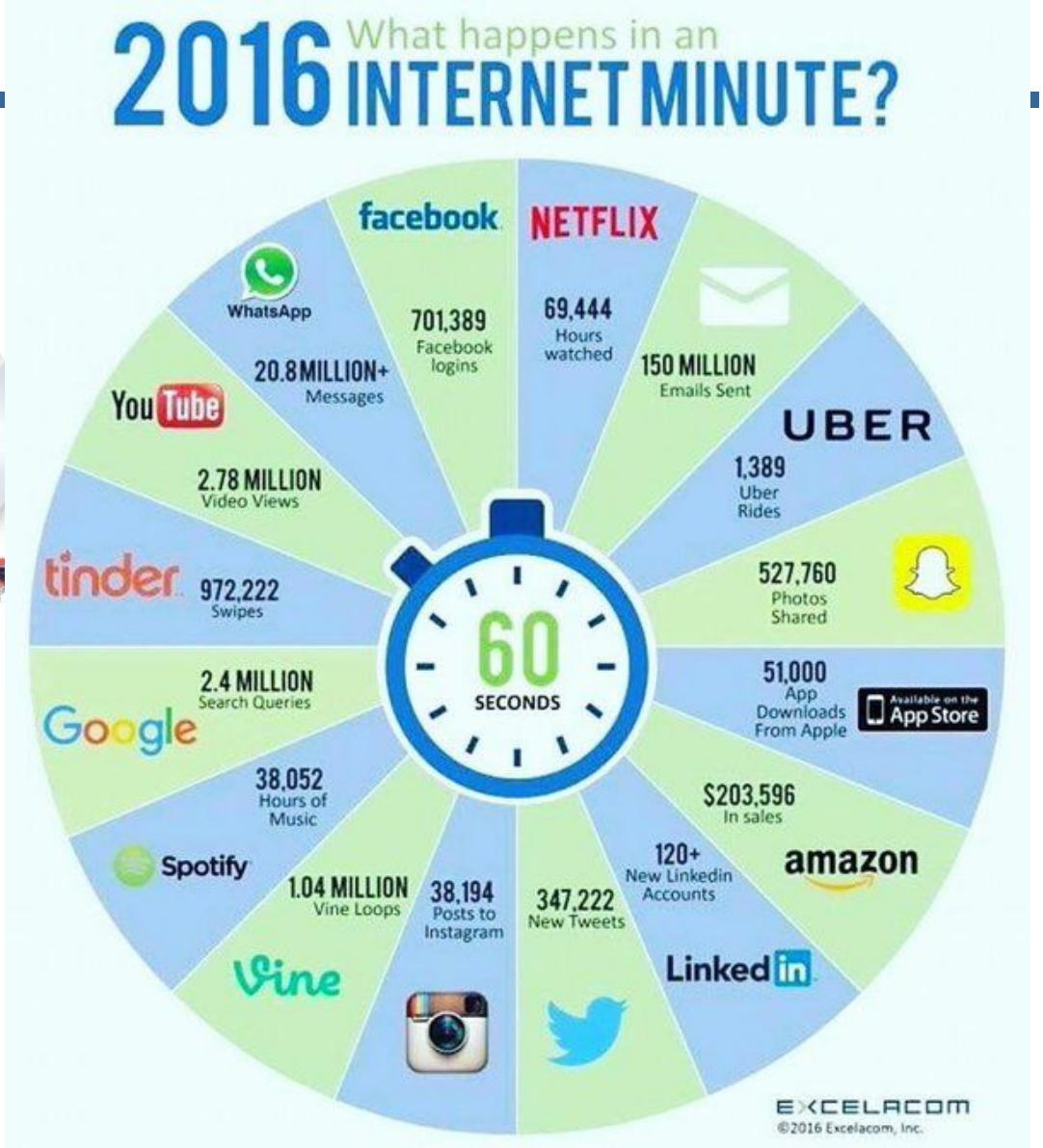
Data Analytics

- Clustering
- Classification
- Regression
- Network analysis
- Visual analytics
- Association
- …

Índice

- **Big Data. Big Data Science**
- **¿Por qué Big Data? Google crea el Modelo de Programación MapReduce**
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

¿Por qué Big Data?



http://www.dailyinfographic.com/what-happens-in-an-internet-minute-infographic

¿Por qué Big Data?

- **Problema:** Escalabilidad de grandes cantidades de datos
- **Ejemplo:**
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos
- **Solución → Divide-Y-Vencerás**

Una sola máquina no puede gestionar grandes volúmenes de datos de manera eficiente

¿Por qué Big Data?

- **Problema:** Escalabilidad de grandes cantidades de datos
- **Ejemplo:**
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos
- **Solución → Divide-Y-Vencerás**

¿Cómo podemos procesar

1000 TB or 10000 TB?

¿Por qué Big Data?

- Escalabilidad de grandes cantidades de datos
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos

Solución → Divide-Y-Vencerás

¿Qué ocurre cuando el tamaño de los datos aumenta y los requerimientos de tiempo se mantiene?

Hace unos años: Había que aumentar los recursos de hardware (número de nodos). Esto tiene limitaciones de espacio, costes, …

Google 2004: Paradigma **MapReduce**

MapReduce

- Escalabilidad de grandes cantidades de datos
  - Exploración 100 TB en 1 nodo @ 50 MB/sec = 23 días
  - Exploración en un clúster de 1000 nodos = 33 minutos

Solución → Divide-Y-Vencerás

MapReduce

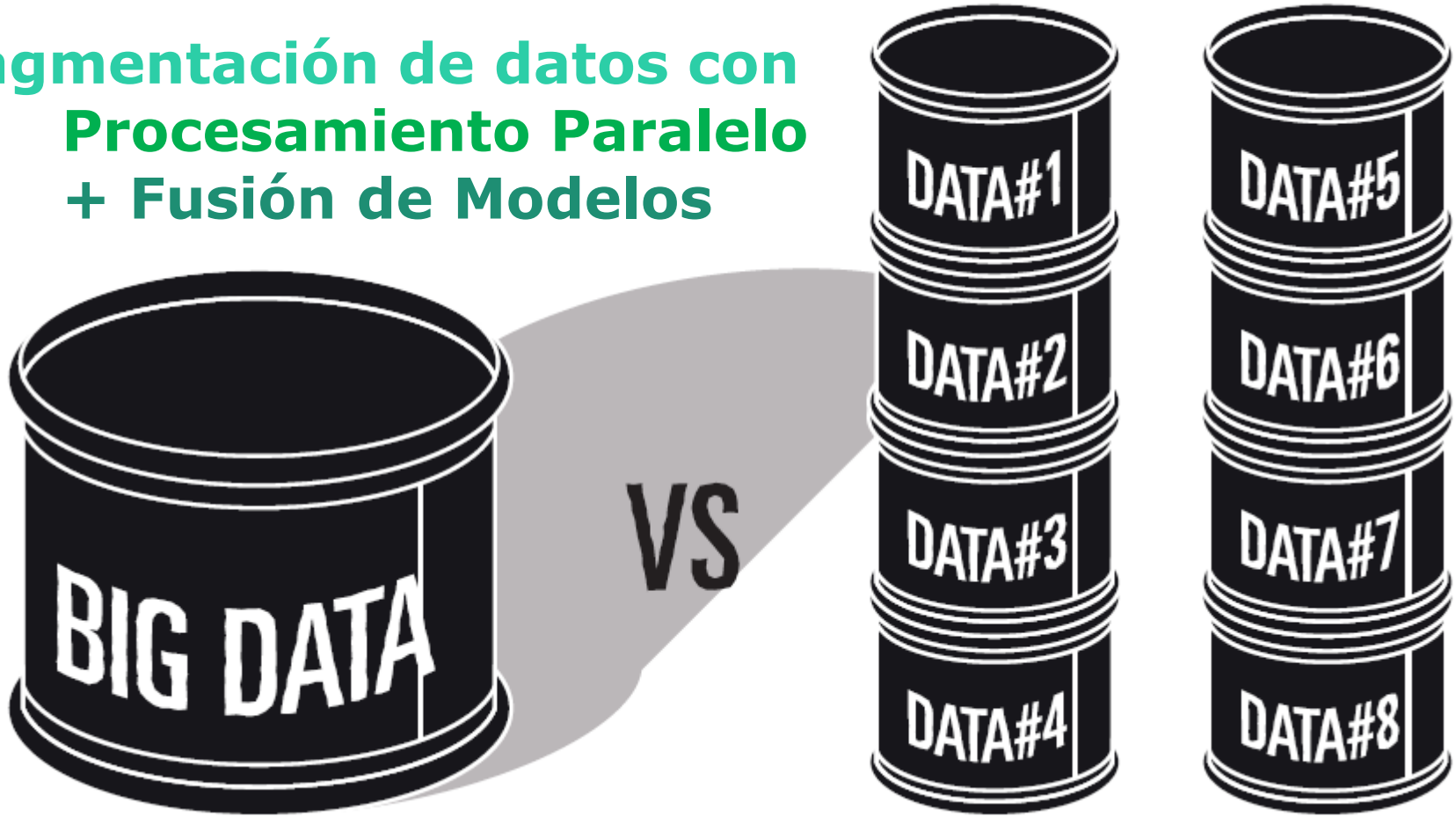
- Modelo de programación de datos paralela
- Concepto simple, elegante, extensible para múltiples aplicaciones

- **Creado por Google (2004)**
  - Procesa 20 PB de datos por día (2004)
- **Popularizado por el proyecto de codigo abierto Hadoop**
  - Usado por Yahoo!, Facebook, Amazon, …

MapReduce

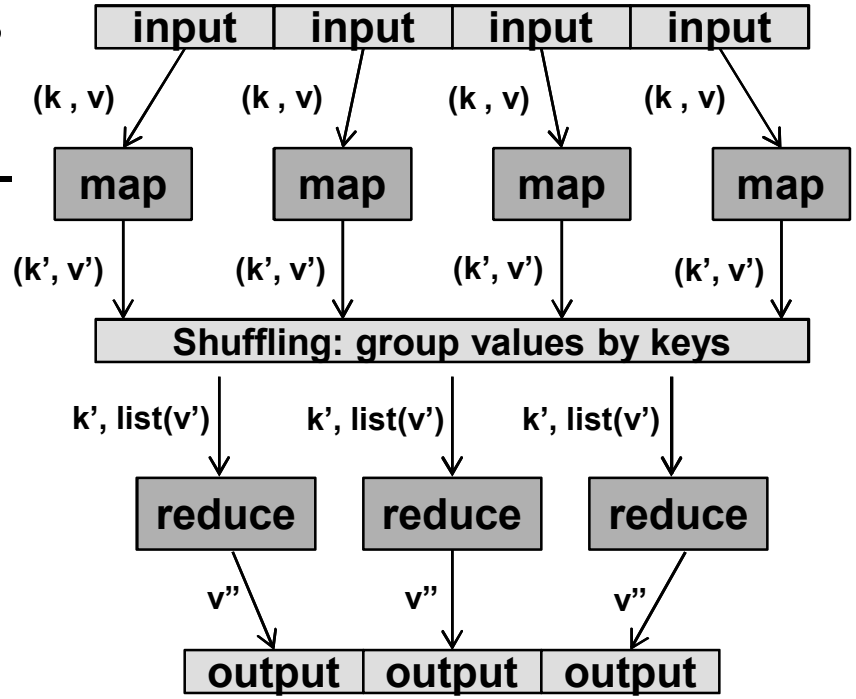
MapReduce es la aproximación más popular para Big Data

Fragmentación de datos con
Procesamiento Paralelo
+ Fusión de Modelos



MapReduce

- MapReduce es el entorno más popular para Big Data
- Basado en la estructura Valor-llave.
- Dos operaciones:
  1. **Función Map : Procesa bloques de información**
  2. **Función Reduce function: Fusiona los resultados previous de acuerdo a su llave.**
- + Una etapa intermedia de agrupamiento por llave **(Shuffling)**



map (k, v) → list (k', v')
reduce (k', list(v')) → v''

J. Dean, S. Ghemawat, MapReduce: Simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107-113.

MapReduce

Resumiendo:

- **Ventaja frente a los modelos distribuidos clásicos:** El modelo de programación paralela de datos de MapReduce oculta la complejidad de la distribución y tolerancia a fallos.
- **Claves de su filosofía**: Es
  - ***escalable:*** *se olvidan los problemas de hardware*
  - ***más barato:*** *se ahorran costes en hardware, programación y administración (Commodity computing).*
- **MapReduce no es adecuado para todos los problemas, pero cuando funciona, puede ahorrar mucho tiempo**

Bibliografía: A. Fernandez, S. Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, **Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks.** *WIREs Data Mining and Knowledge Discovery 4:5 (2014) 380-409*

MapReduce

Limitaciones

"If all you have is a hammer, then everything looks like a nail."

Los siguientes tipos de algoritmos son ejemplos en los que MapReduce no funciona bien:

Iterative Graph Algorithms
Gradient Descent
Expectation Maximization

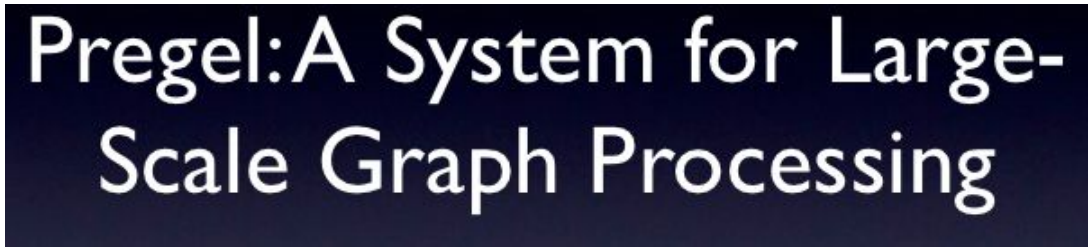
Limitaciones de MapReduce

Algoritmos de grafos iterativos. Existen muchas limitaciones para estos algoritmos.

Ejemplo: Cada iteración de PageRank se corresponde a un trabajo de MapReduce.

Se han propuesto una serie de extensiones de MapReduce o modelos de programación alternativa para acelerar el cálculo iterativo:

Pregel (Google)



Implementación: http://www.michaelnielsen.org/ddi/pregel/

Malewicz, G., Austern, M., Bik, A., Dehnert, J., Horn, I., Leiser, N., and Czajkowski, G. Pregel: A system for large escale graph processing. ACM SIGMOD 2010.

Limitaciones de MapReduce

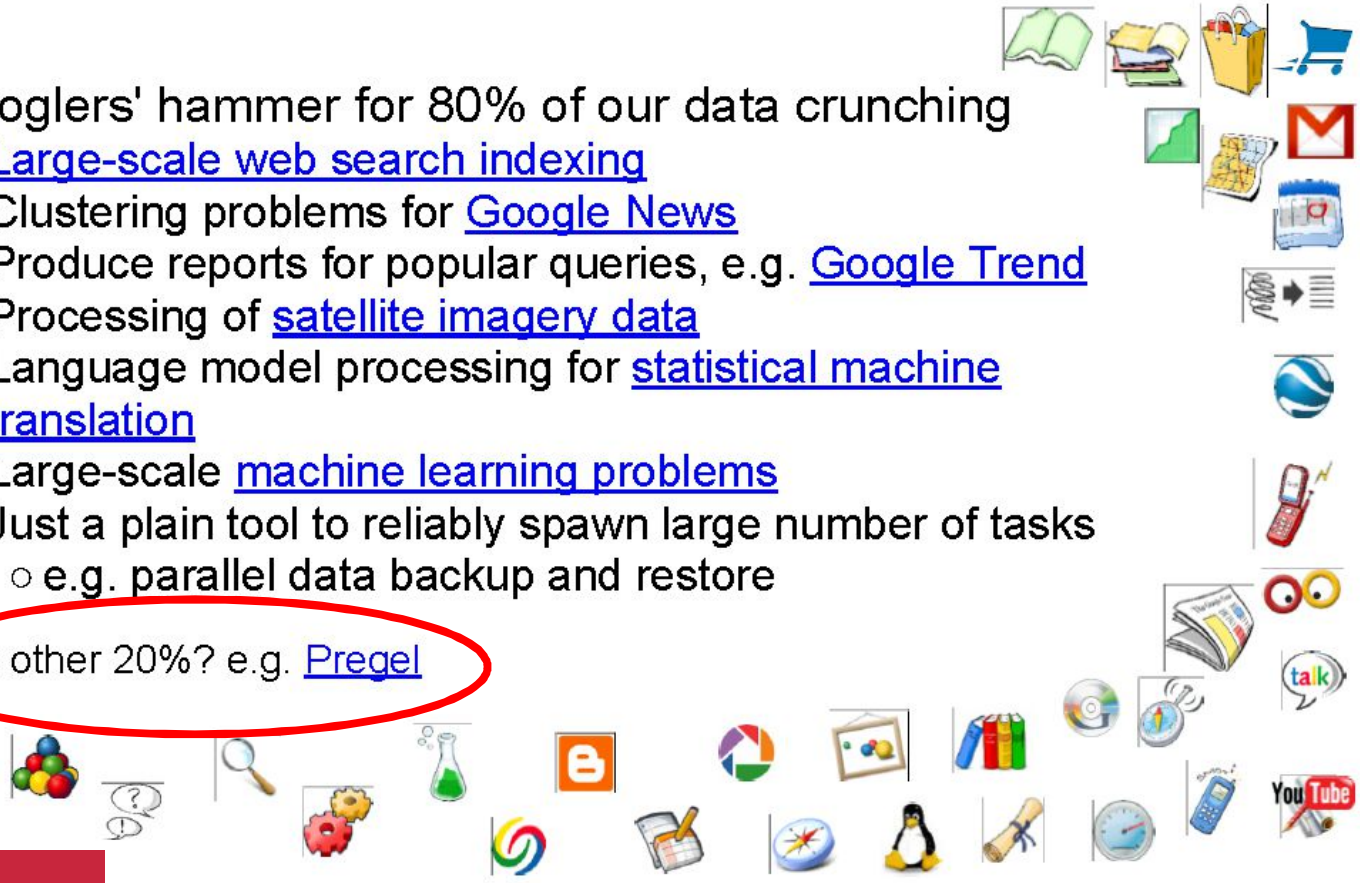
MapReduce inside Google

Google

Googlers' hammer for 80% of our data crunching

- Large-scale web search indexing
- Clustering problems for Google News
- Produce reports for popular queries, e.g. Google Trend
- Processing of satellite imagery data
- Language model processing for statistical machine translation
- Large-scale machine learning problems
- Just a plain tool to reliably spawn large number of tasks
  - e.g. parallel data backup and restore

The other 20%? e.g. Pregel



Enrique Alfonseca
Google Research Zurich

Limitaciones de MapReduce

En Resumen

Principales características

- Arquitectura escalable
- Planificación optimizada
- Elasticidad y disponibilidad
- Flexibilidad
- Seguridad y Autenticación

Limitaciones

- Aprendizaje automático: Computación iterativa
- Procesamiento de grafos
- Procesamiento en tiempo real (streams)

Bibliografía: A. Fernandez, S. Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, **Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks.** *WIREs Data Mining and Knowledge Discovery 4:5 (2014) 380-409*

Índice

- **Big Data. Big Data Science**
- **¿Por qué Big Data? Google crea el Modelo de Programación MapReduce**
- **Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)**
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

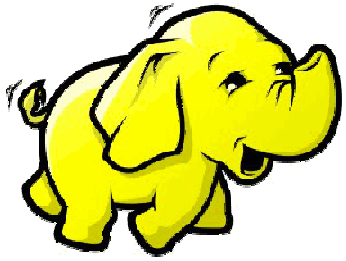
Hadoop

Hadoop es una implementación de código abierto del paradigma computacional MapReduce

http://hadoop.apache.org/

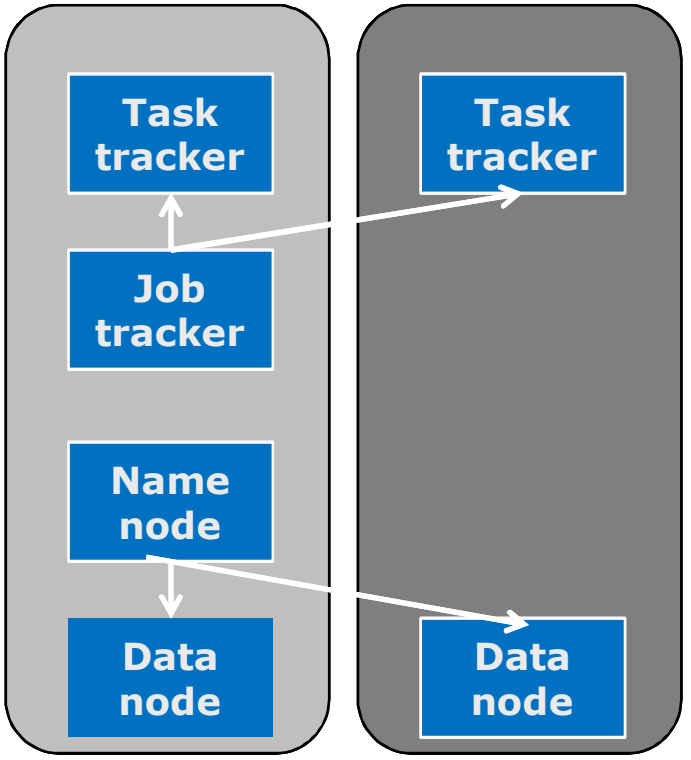
Hadoop

Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido, escalable y portátil escrito en Java para el framework Hadoop



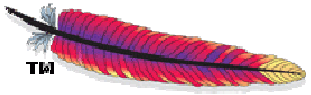
Map Reduce Layer

HDFS Layer



Creado por Doug Cutting (chairman of board of directors of the Apache Software Foundation, 2010)

http://hadoop.apache.org/



Ecosistema Hadoop

El proyecto Apache Hadoop incluye los módulos:

Hadoop Common: Las utilidades comunes que apoyan los otros módulos de Hadoop.
Hadoop Distributes File System (HDFS): El sistema de ficheros que proporciona el acceso
Hadoop YARN: Marco para el manejo de recursos de programación y grupo de trabajo.
Hadoop MapReduce: Un sistema de basado en YARN o para el procesamiento en paralelo de grandes conjuntos de datos.

http://hadoop.apache.org/

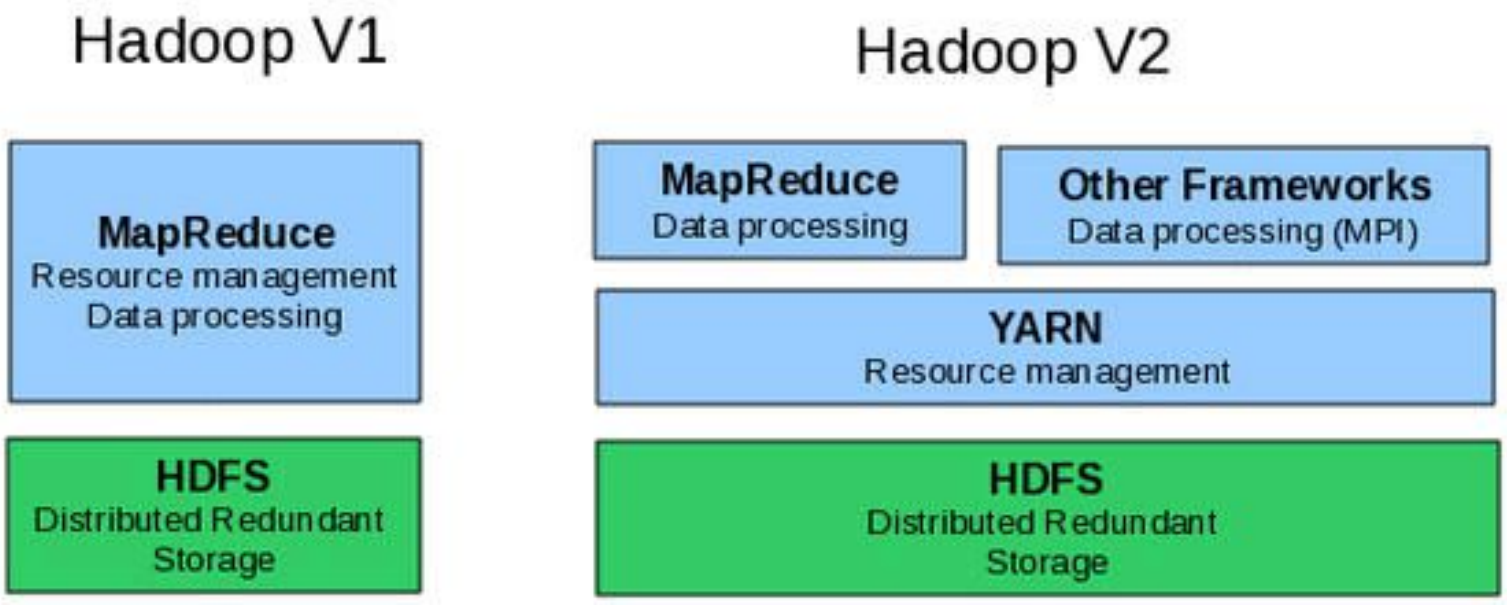
Ecosistema Apache Hadoop incluye más de 150 proyectos:

Avro: Un sistema de serialización de datos.
Cassandra: Una base de datos escalable multi-master sin puntos individuales y fallo
Chukwa: Un sistema de recogida de datos para la gestión de grandes sistemas distribuidos.
Hbase: Una base de datos distribuida, escalable que soporta estructurado de almacenamiento de datos para tablas de gran tamaño.
Hive: Un almacén de datos que proporciona el Resumen de datos para tablas de gran tamaño.
Pig: Lenguaje para la ejecución de alto nivel de flujo de datos para computación paralela.
Tez: Sustituye al modelo "MapShuffleReduce" por un flujo de ejecución con grafos acíclico dirigido (DAG)
Giraph: Procesamiento iterativo de grafos
Mahout: Aprendizaje automático escalable (biblioteca de minería de datos)

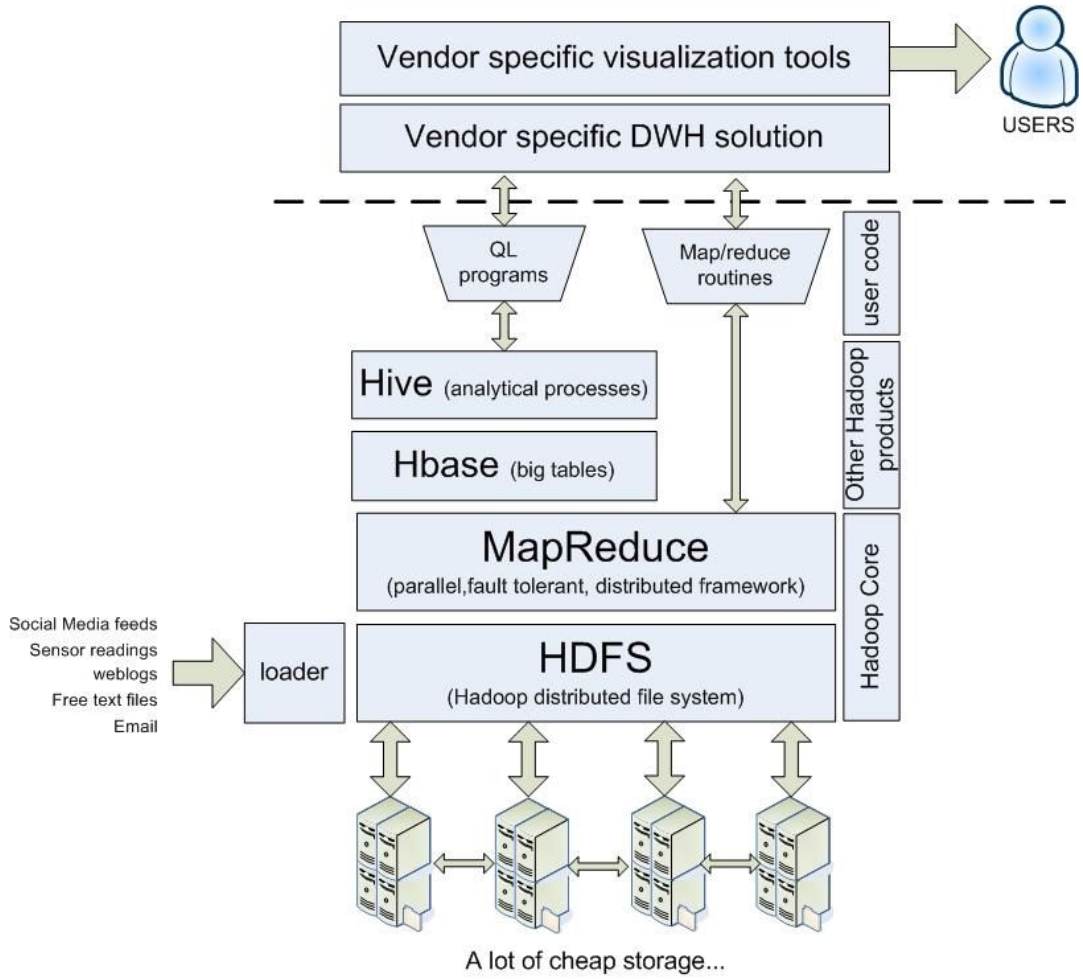
Recientemente: Apache Spark

Evolución de Hadoop

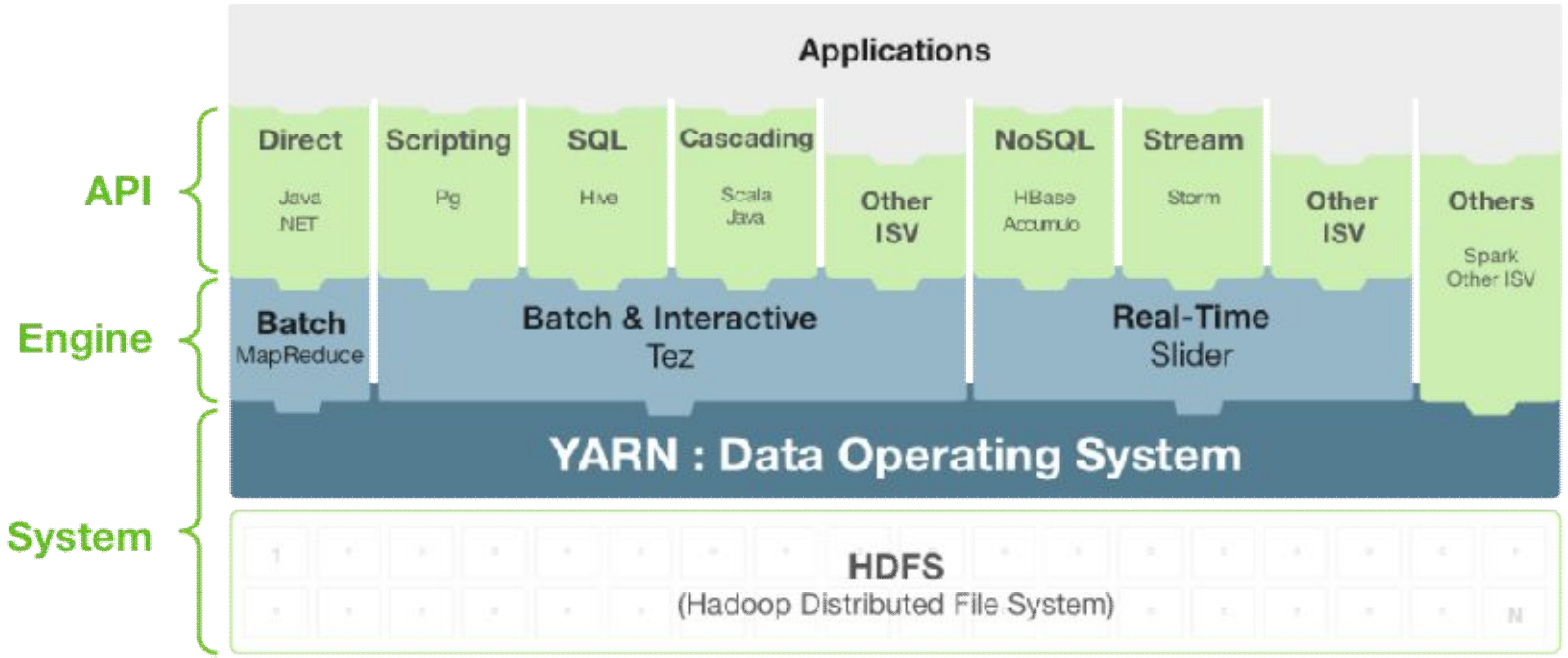
Evolución de Hadoop



Evolución de Hadoop

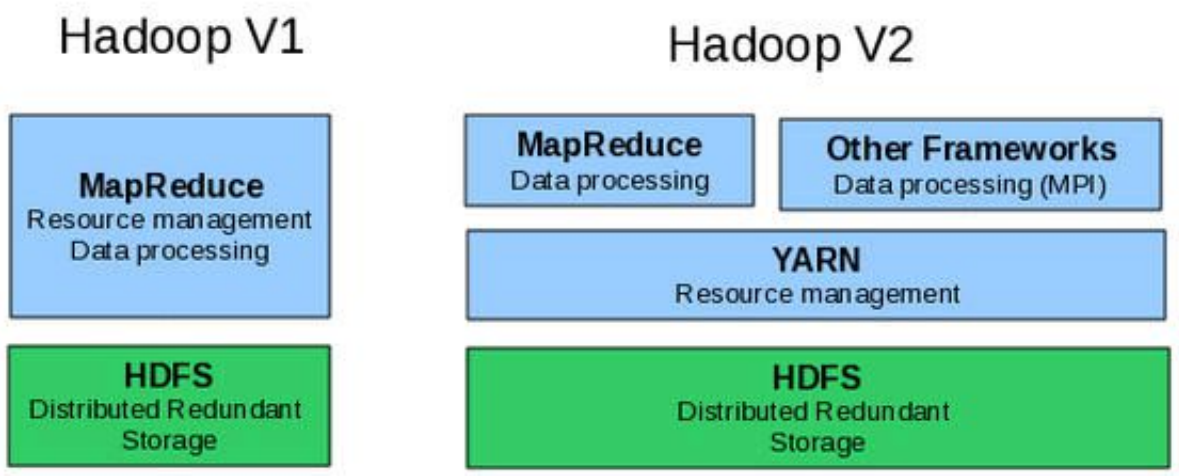


Apache Hadoop YARN



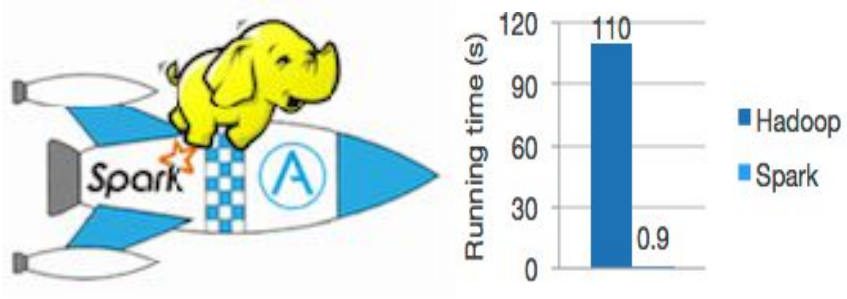
Apache Hadoop YARN es el sistema operativo de datos de Hadoop 2, responsable de la gestión del acceso a los recursos críticos de Hadoop. YARN permite al usuario interactuar con todos los datos de múltiples maneras al mismo tiempo, haciendo de Hadoop una verdadera plataforma de datos multi-uso y lo que le permite tomar su lugar en una arquitectura de datos moderna.

Apache Spark



https://spark.apache.org/

Enfoque InMemory
HDFS Hadoop + SPARK



Fast and expressive cluster computing system compatible with Apache Hadoop

Apache Spark (Birth 2009-2010)

Fast and Expressive Cluster Computing Engine Compatible with Apache Hadoop

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

2-5x less code

Efficient

- General execution graphs
- In-memory storage

Usable

- Rich APIs in Java, Scala, Python
- Interactive shell

http://spark.apache.org/talks/pydata-july-2013.pdf

Apache Spark

Spark Programming Model

KEY Concept: RDD (Resilient Distributed Datasets)

Write programs in terms of operations on distributed data sets

- Collection of objects spread across a cluster, stored in RAM or on Disk
- Built through parallel transformations on distributed datasets
- An RDD is a fault-tolerant collection of elements that can be operated on in parallel.
- There are two ways to create RDDs:

  Parallelizing an existing collection in your driver program

  Referencing a dataset in an external storage system, such as a shared filesystem, HDFS, Hbase.
- *Can be cached for future reuse*
- Built through parallel transformations on distributed datasets
- RDD operations: transformations and actions

Transformations (e.g. map, filter, groupBy)…

(Lazy operations to build RDSs from other RDSs)

Actions (eg. Count, collect, save …)

(Return a result or write it a storage)

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I. **Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.** In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

Spark Operations

Transformations (define a new RDD)	map filter sample groupByKey reduceByKey sortByKey	flatMap union join cogroup cross mapValues
Actions (return a result to driver program)	collect reduce count save lookupKey	

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I.
Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.
In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

- RDDs allow us to express different programming models:
  - MapReduce.
  - Iterative MapReduce. It can implement HaLoop or Twister systems.
  - Stream processing.
  - Iterative graph applications (Google's Pregel).
  - SQL.
  - ...
- It provides us more flexibility to design scalable ML tools.

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I.
Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.
In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

DataFrame-based API is primary API
The MLlib RDD-based API is now in maintenance mode.

DataFrames

A DataFrame is a distributed collection of data organized into named columns. It is conceptually equivalent to a table in a relational database or a data frame in R/Python, but with richer optimizations under the hood. DataFrames can be constructed from a wide array of sources such as: structured data files, tables in Hive, external databases, or existing RDDs.

The DataFrame API is available in Scala, Java, Python, and R.

Datasets

A Dataset is a new experimental interface added in Spark 1.6 that tries to provide the benefits of RDDs (strong typing, ability to use powerful lambda functions) with the benefits of Spark SQL's optimized execution engine. A Dataset can be constructed from JVM objects and then manipulated using functional transformations (map, flatMap, filter, etc.).

The unified Dataset API can be used both in Scala and Java. Python does not yet have support for the Dataset API, but due to its dynamic nature many of the benefits are already available (i.e. you can access the field of a row by name naturally `row.columnName`). Full python support will be added in a future release.

Zaharia-2012- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I.
Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing.
In: *9th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, 2012,* 1–14.

Apache Spark

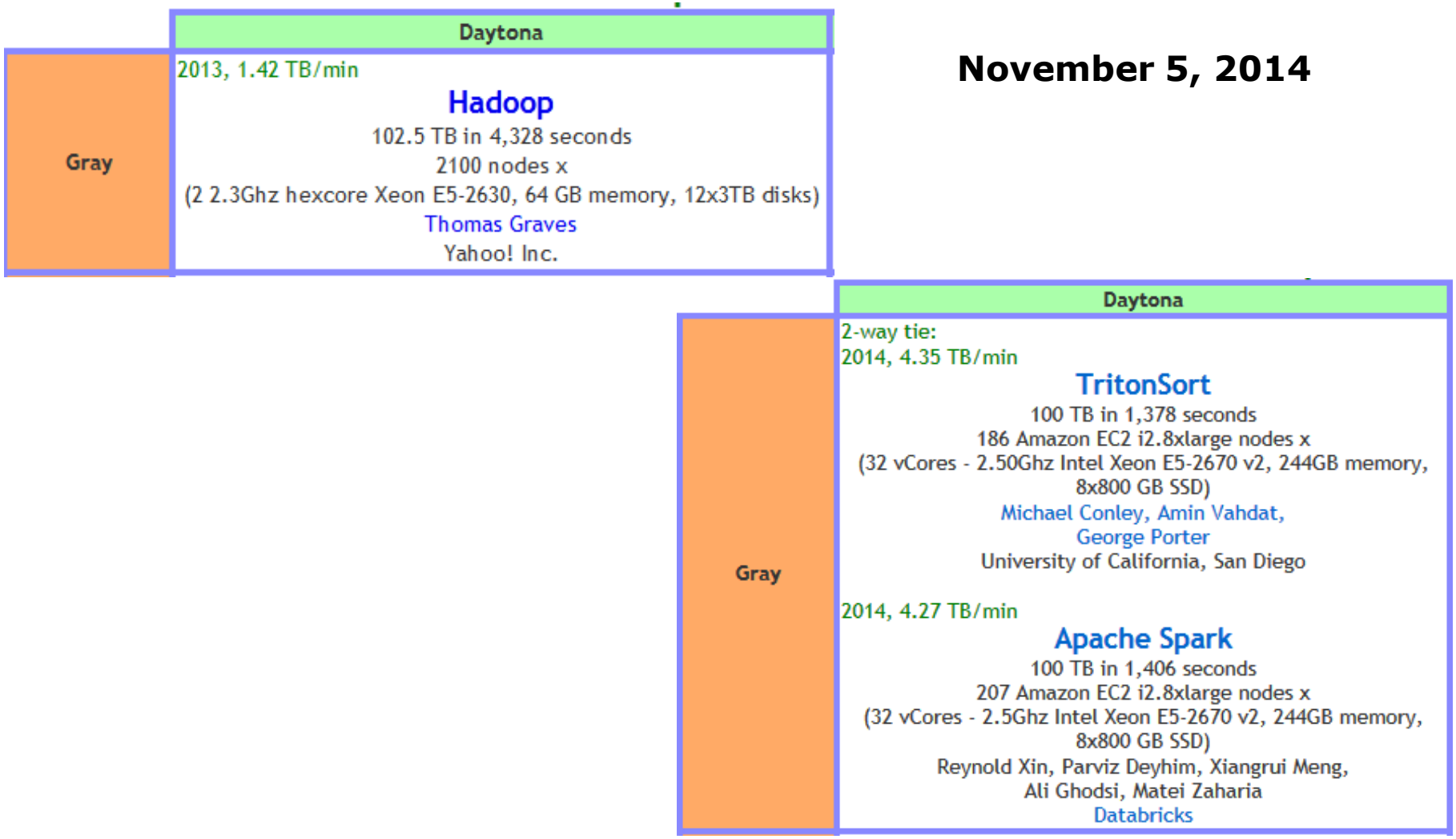
	Hadoop World Record	Spark 100 TB *
Data Size	102.5 TB	100 TB
Elapsed Time	72 mins	23 mins
# Nodes	2100	206
# Cores	50400	6592
# Reducers	10,000	29,000
Rate	1.42 TB/min	4.27 TB/min
Rate/node	0.67 GB/min	20.7 GB/min
Sort Benchmark Daytona Rules	Yes	Yes
Environment	dedicated data center	EC2 (i2.8xlarge)

October 10, 2014

Using Spark on 206 EC2 nodes, we completed the benchmark in 23 minutes. This means that Spark sorted the same data 3X faster using 10X fewer machines. All the sorting took place on disk (HDFS), without using Spark's in-memory cache.

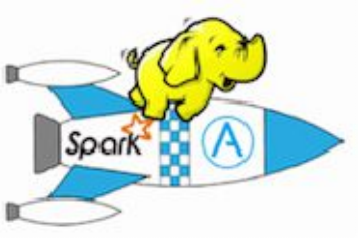
http://databricks.com/blog/2014/10/10/spark-petabyte-sort.html

Apache Spark

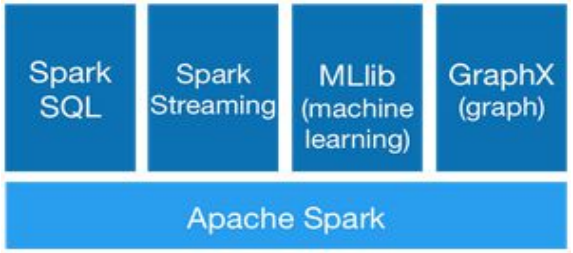


http://sortbenchmark.org/

Apache Spark

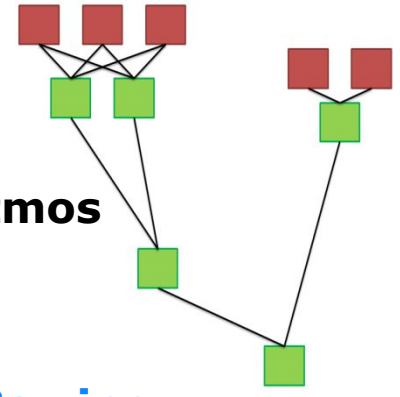


Ecosistema Apache Spark



● **Big Data "in-memory".** Spark permite realizar trabajos paralelizados totalmente en memoria, lo cual reduce mucho los tiempos de procesamiento. Sobre todo si se trata de unos procesos iterativos. En el caso de que algunos datos no quepan en la memoria, Spark seguirá trabajando y usará el disco duro para volcar aquellos datos que no se necesitan en este momento (Hadoop **"commodity hardware")**.

KEY Concept: RDD (Resilient Distributed Datasets)
Write programs in terms of operations on distributed data sets



● **Esquema de computación más flexible que MapReduce.**
Permite la flujos acíclicos de procesamiento de datos, algoritmos iterativos

● **Spark ofrece una API para Java, Python y Scala**

● Spark / Wiki Homepage
Powered By Spark
Creado por Andy Konwinski, modificado por última vez por Reynold Xin el sep 08, 2015

Databricks, Groupon, eBay inc., Amazon, Hitachi, Nokia, Yahoo!, …

https://cwiki.apache.org/confluence/display/SPARK/Powered+By+Spark

Flink

https://flink.apache.org/

Flink | Overview | Features | Downloads | FAQ | Quickstart | Documentation

Apache Flink is an open source platform for distributed stream and batch data processing.

Flink's core is a streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams.

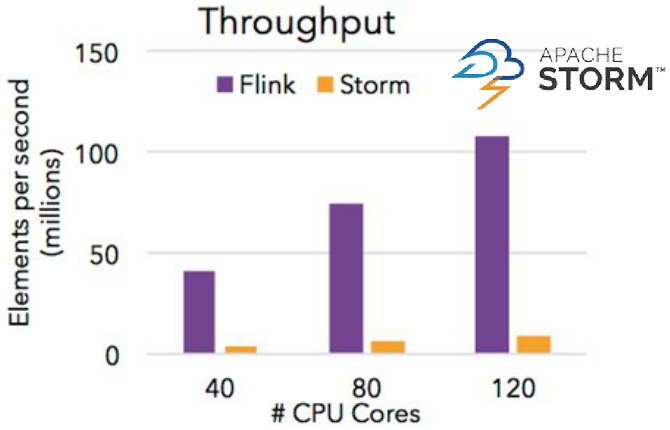
Flink includes **several APIs** for creating applications that use the Flink engine:

1. DataStream API for unbounded streams embedded in Java and Scala, and
2. DataSet API for static data embedded in Java, Scala, and Python,
3. Table API with a SQL-like expression language embedded in Java and Scala.

Flink also bundles **libraries for domain-specific use cases**:

1. CEP, a complex event processing library,
2. Machine Learning library, and
3. Gelly, a graph processing API and library.

You can **integrate** Flink easily with other well-known open source systems both for data input and output as well as deployment.



Batch on Streaming

Batch processing applications run efficiently as special cases of stream processing applications.

Streaming First

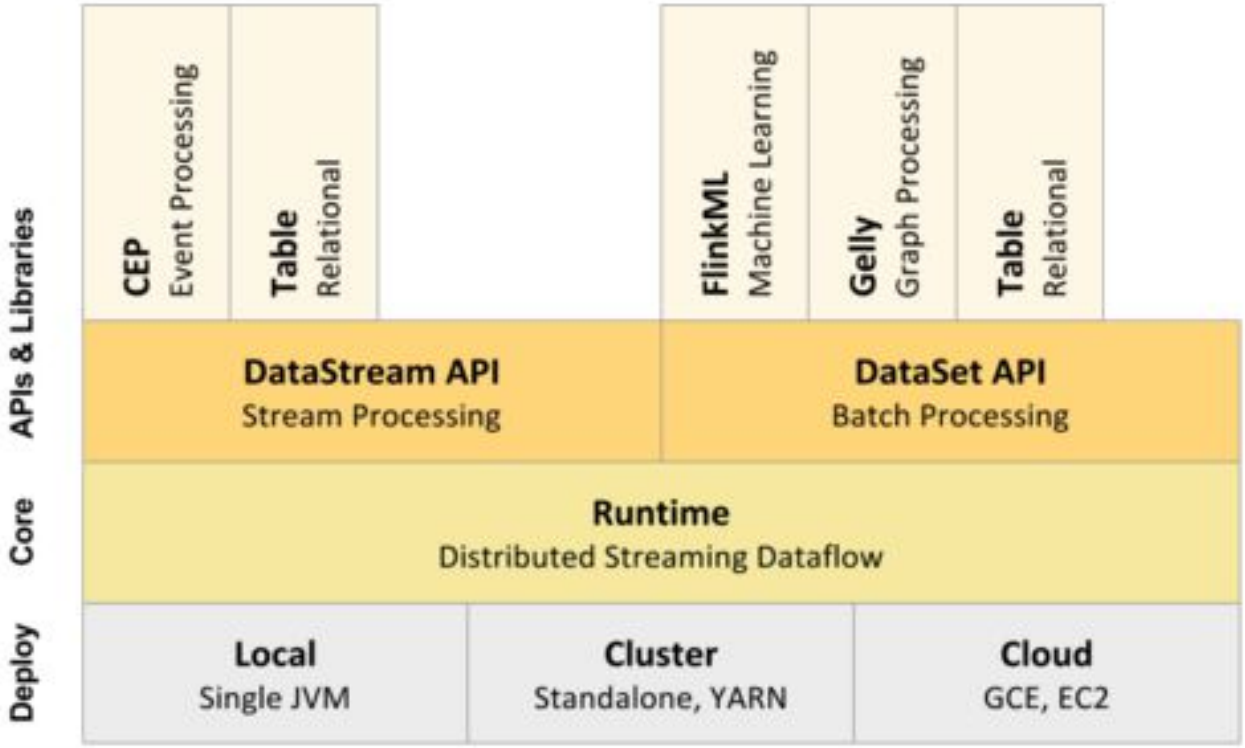
High throughput and low latency stream processing with exactly-once guarantees.

APIs, Libraries, and Ecosystem

DataSet, DataStream, and more. Integrated with the Apache Big Data stack.

Flink

https://flink.apache.org/



Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …) (Una instantánea)

Big Data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.

There are two main issues related to Big Data:

1. **Database/storage frameworks: to write, read, and manage data.**
2. **Computational models: to process and analyze data.**

Recently, there are the following Big Data frameworks:

1. **Storage frameworks: Google File System (GFS), Hadoop Distributed File Systems (HDFS).**
2. **Computational models: MapReduce (Apache Hadoop), Resilient Distributed Datasets (RDD by Apache Spark, DataFrames, Dataset API (Spark 1.6)).**

2001-2010

2010-2018

Big Data: Technology and Chronology

The Apache Software Foundation

BIG DATA

2001

3V's Gartner

Doug Laney

Google

Hadoop Ecosystem

2009-2013 Flink

TU Berlin

Flink Apache (Dec. 2014) Volker Markl

2004

MapReduce

Google

Jeffrey Dean

Flink

Big Data

YAHOO!

hadoop

2010 Spark

U Berckeley

Apache Spark

Feb. 2014

Matei Zaharia

2008

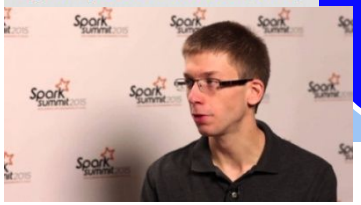
Hadoop

Yahoo!

Doug Cutting

Spark

Lightning-Fast Cluster Computing



2010-2018:

Big Data Analytics: Mahout, MLLib, …

Hadoop Ecosystem

Applications New Technology

Índice

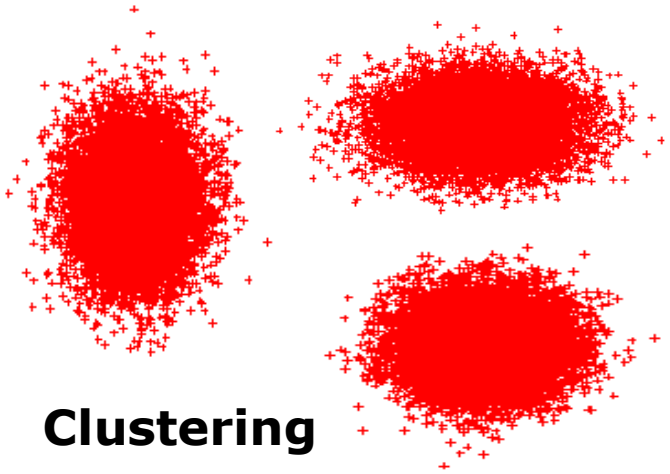
- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- **Big Data Analytics: Librerías para Analítica de Datos en Big Data.**
- Casos de estudio: Random Forest
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

Big Data Analytics

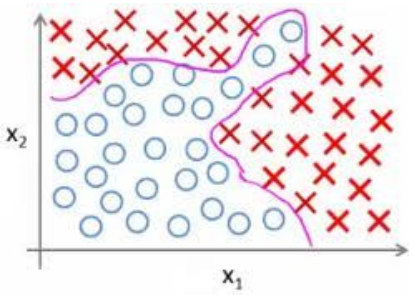
- **Big Data Analytics: Escenario**
- **Big Data Analytics: Tools**
  (Mahout, MLlib, FlinkML, H2O)
- **Caso de estudio: Random Forest**
- **Big Data Analytics: 3 Comentarios finales.**
  - Without Analytics, Big data is just noise: Smart Data
  - Big data preprocessing: Es necesario
  - Los expertos en Ciencia de Datos son necesarios en el uso de herramientas de Analytics y Big Data.

Big Data Analytics

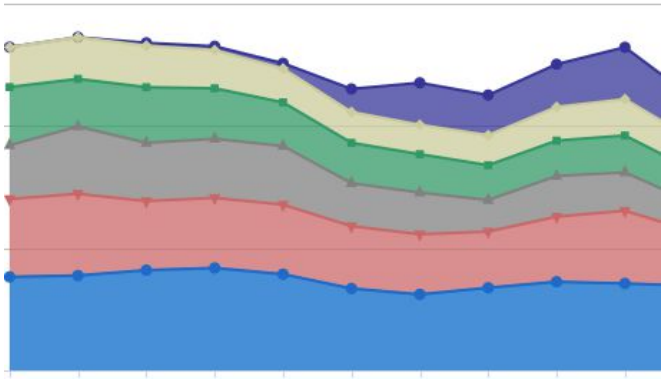
Potentiales escenarios:



Clustering

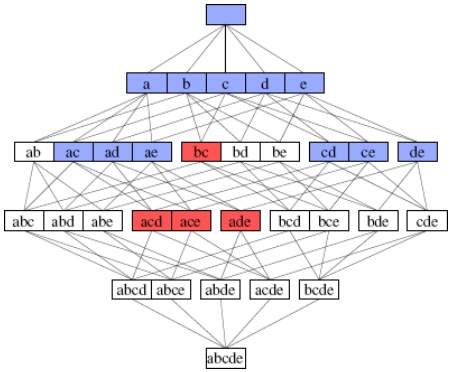


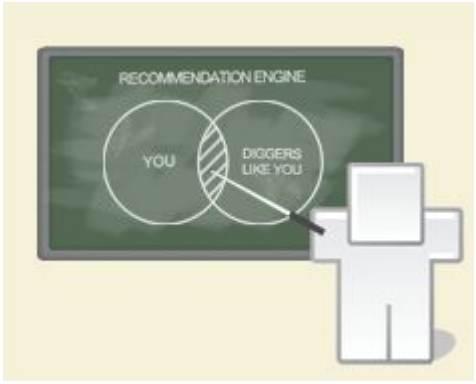
Classification



Real Time Analytics/ Big Data Streams

Association





Recommendation Systems

Social Media Mining

Social Big Data

Big Data Analytics: **Tools**

Generation	1st Generation	2nd Generation	3nd Generation
Examples	SAS, R, Weka, SPSS, KEEL	Mahout, Pentaho, Cascading	Spark, Haloop, GraphLab, Pregel, Giraph, ML over Storm
Scalability	Vertical	Horizontal (over Hadoop)	Horizontal (Beyond Hadoop)
Algorithms Available	Huge collection of algorithms	Small subset: sequential logistic regression, linear SVMs, Stochastic Gradient Decendent, k-means clustsering, Random forest, etc.	Much wider: CGD, ALS, collaborative filtering, kernel SVM, matrix factorization, Gibbs sampling, etc.
Algorithms Not Available	Practically nothing	Vast no.: Kernel SVMs, Multivariate Logistic Regression, Conjugate Gradient Descendent, ALS, etc.	Multivariate logistic regression in general form, k-means clustering, etc. – Work in progress to expand the set of available algorithms
Fault-Tolerance	Single point of failure	Most tools are FT, as they are built on top of Hadoop	FT: HaLoop, Spark Not FT: Pregel, GraphLab, Giraph

Big Data Analytics: Tools

Mahout

Classification	Single Machine	MapReduce
Logistic Regression - trained via SGD	x	
Naive Bayes / Complementary Naive Bayes		x
Random Forest		x
Hidden Markov Models	x	
Multilayer Perceptron	x	

Mahout Samsara

MLlib

MLlib types, algorithms and utilities

This lists functionality included in `spark.mllib`, the main MLlib API.

- Data types
- Basic statistics
  - summary statistics
  - correlations
  - stratified sampling
  - hypothesis testing
  - random data generation
- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)
- PMML model export

https://spark.apache.org/mllib/

Spark Libraries

https://spark.apache.org/

APACHE Spark™ *Lightning-fast cluster computing*

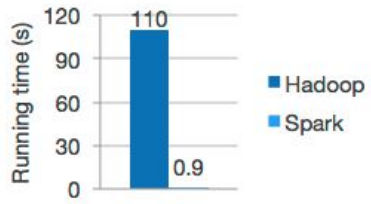
Download | Libraries | Documentation | Examples | Community | Developers | Apache Software Foundation

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.

Apache Spark has an advanced DAG execution engine that supports acyclic 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

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

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

Latest News

Spark+AI Summit (June 4-6th, 2018, San Francisco) agenda posted (Mar 01, 2018)

Spark 2.3.0 released (Feb 28, 2018)

Spark 2.2.1 released (Dec 01, 2017)

Spark 2.1.2 released (Oct 09, 2017)

Archive

Download Spark

Built-in Libraries:

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

Third-Party Projects

Spark Libraries

https://spark.apache.org/

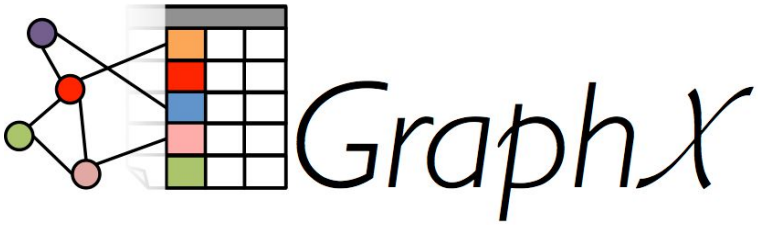
Lightning-fast cluster computing

Download | Libraries | Documentation | Examples | Community | FAQ

- APIs: RDD, DataFrame and SQL
- Backend Execution: DataFrame and SQL
- Integrations: Data Sources, Hive, Hadoop, Mesos and Cluster Management
- R Language
- Machine Learning and Advanced Analytics
- Spark Streaming
- Deprecations, Removals, Configs, and Behavior Changes
  - Spark Core
  - Spark SQL & DataFrames
  - Spark Streaming
  - MLlib
- Known Issues
  - SQL/DataFrame
  - Streaming
- Credits

MLlib, Streaming and GraphX

https://spark.apache.org/streaming/



https://spark.apache.org/graphx/

MLlib

Machine Learning Library (MLlib) Guide

MLlib is Spark's machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. At a high level, it provides tools such as:

- ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
- Persistence: saving and load algorithms, models, and Pipelines
- Utilities: linear algebra, statistics, data handling, etc.

https://spark.apache.org/docs/latest/ml-guide.html

MLlib and Spark Packages

https://spark.apache.org/docs/latest/ml-guide.html

Spark MLlib

MLlib: Main Guide

- Basic statistics
- Pipelines
- Extracting, transforming and selecting features
- Classification and Regression
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

«

MLlib: RDD-based API Guide

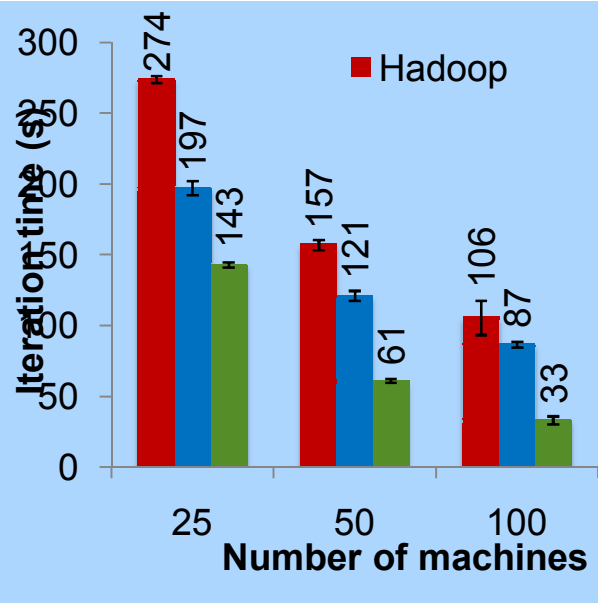
- Data types
- Basic statistics
- Classification and regression
- Collaborative filtering
- Clustering
- Dimensionality reduction
- Feature extraction and transformation
- Frequent pattern mining
- Evaluation metrics
- PMML model export
- Optimization (developer)

http://spark-packages.org/

SparkPackages

A community index of packages for Apache Spark.

K-Means



[Zaharia et. al, NSDI'12]

MLlib

https://spark.apache.org/docs/latest/mllib-guide.html

MLlib: RDD-based API

This page documents sections of the MLlib guide for the RDD-based API (the `spark.mllib` package). Please see the MLlib Main Guide for the DataFrame-based API (the `spark.ml` package), which is now the primary API for MLlib.

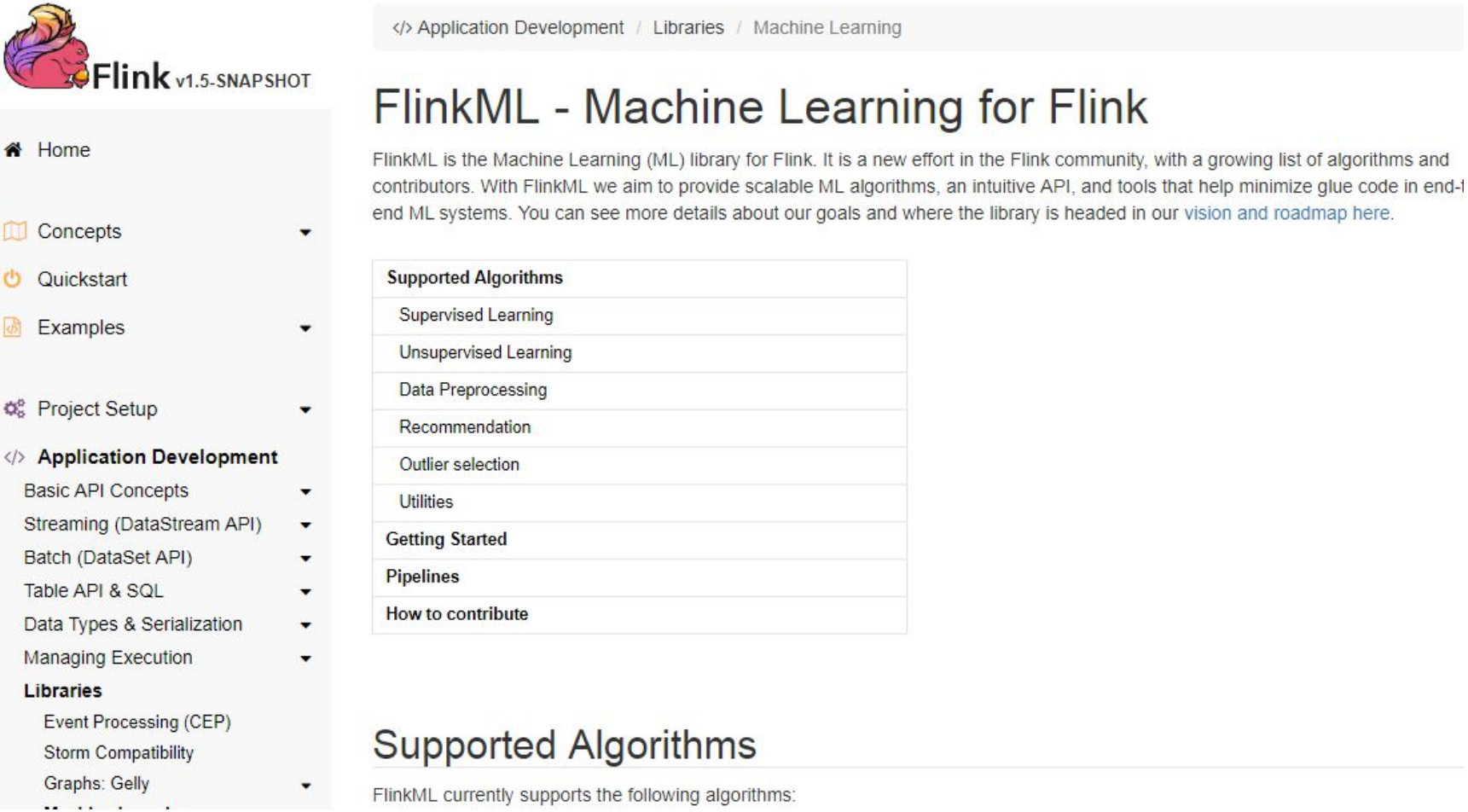
- Data types
- Basic statistics
  - summary statistics
  - correlations
  - stratified sampling
  - hypothesis testing
  - streaming significance testing
  - random data generation
- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - bisecting k-means
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
  - association rules
  - PrefixSpan
- Evaluation metrics
- PMML model export
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)

http://spark-packages.org/

A community index of packages for Apache Spark.

FlinkML

https://ci.apache.org/projects/flink/flink-docs-master/dev/libs/ml/index.html



FlinkML

https://ci.apache.org/projects/flink/flink-docs-master/dev/libs/ml/index.html

Supported Algorithms

FlinkML currently supports the following algorithms:

Supervised Learning

- SVM using Communication efficient distributed dual coordinate ascent (CoCoA)
- Multiple linear regression
- Optimization Framework

Unsupervised Learning

- k-Nearest neighbors join

Data Preprocessing

- Polynomial Features
- Standard Scaler
- MinMax Scaler

Recommendation

- Alternating Least Squares (ALS)

Outlier selection

- Stochastic Outlier Selection (SOS)

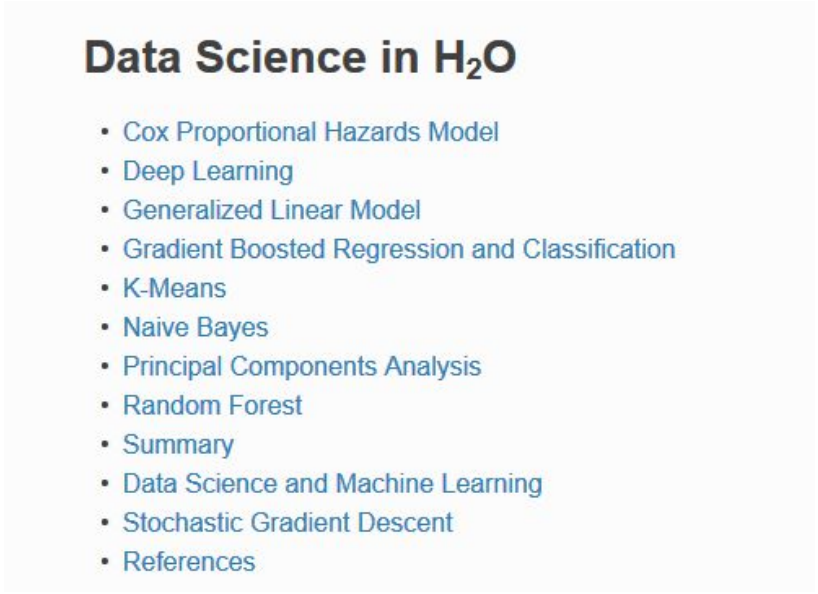
Utilities

- Distance Metrics
- Cross Validation

Librería H_2O

H_2O

H_2O **https://www.h2o.ai/**



Soporte para R, Python, Hadoop y Spark

Funcionamiento: Crea una máquina virtual con Java en la que optimiza el paralelismo de los algoritmos

https://www.h2o.ai/

Librería H_2O

H_2O

http://www.h2o.ai/resources/

H_2O APIs

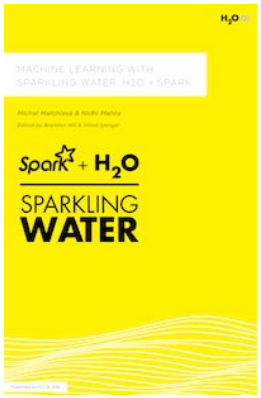
Overview and walkthroughs for the different APIs to H_2O.

- R On H_2O
- Tableau on H_2O

Spark + H_2O SPARKLING WATER

http://h2o-release.s3.amazonaws.com/h2o/rel-turan/4/docs-website/h2o-r/h2o_package.pdf

Machine Learning with Sparkling Water: H2O + Spark

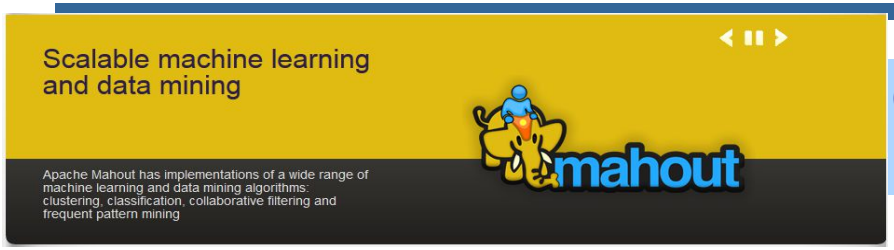


Sparkling Water allows users to combine the fast, scalable machine learning algorithms of H2O with the capabilities of Spark. With Sparkling Water, users can drive computation from Scala/R/Python and utilize the H2O Flow UI, providing an ideal machine learning platform for application developers.

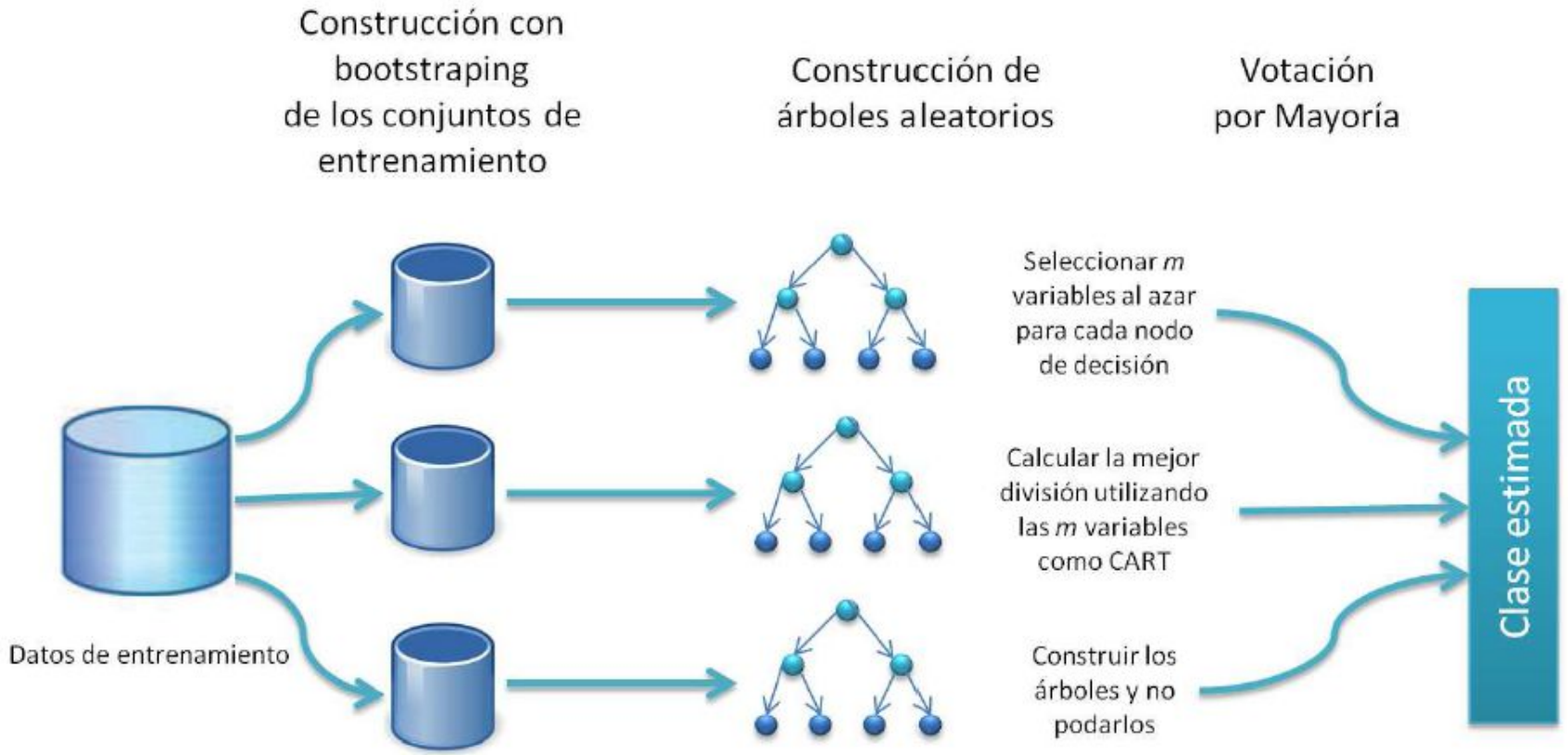
Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- **Casos de estudio: Random Forest.**
  Big Data Analytics: Consideraciones
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

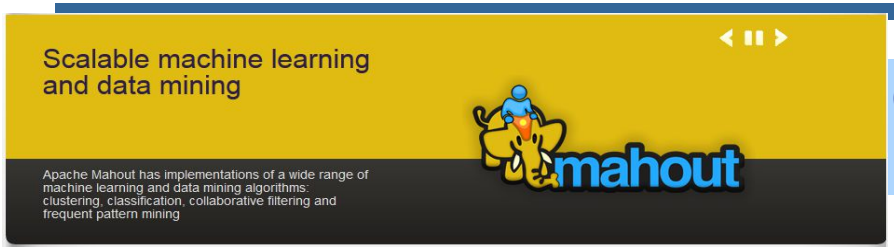
Mahout. Caso de estudio



Caso de estudio:
Random Forest para KddCup99



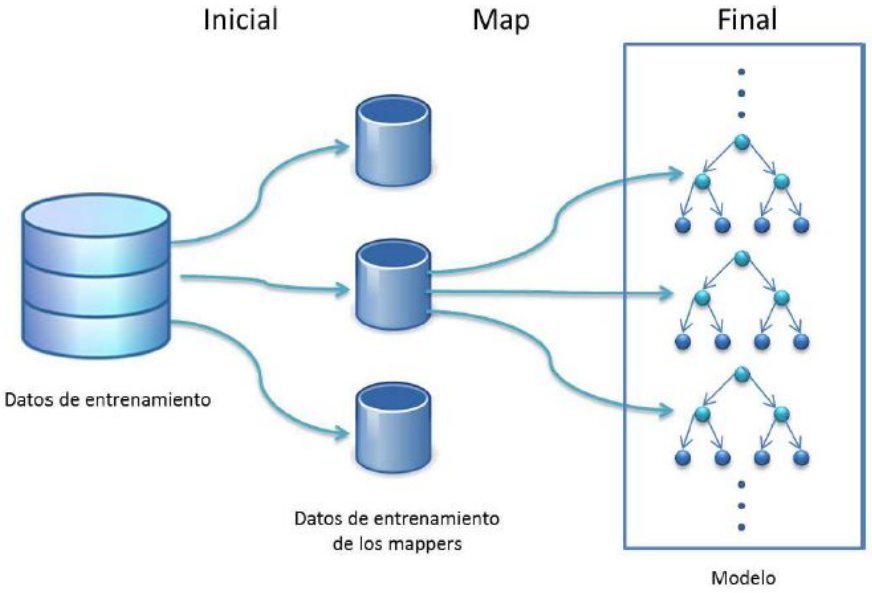
Mahout. Caso de estudio



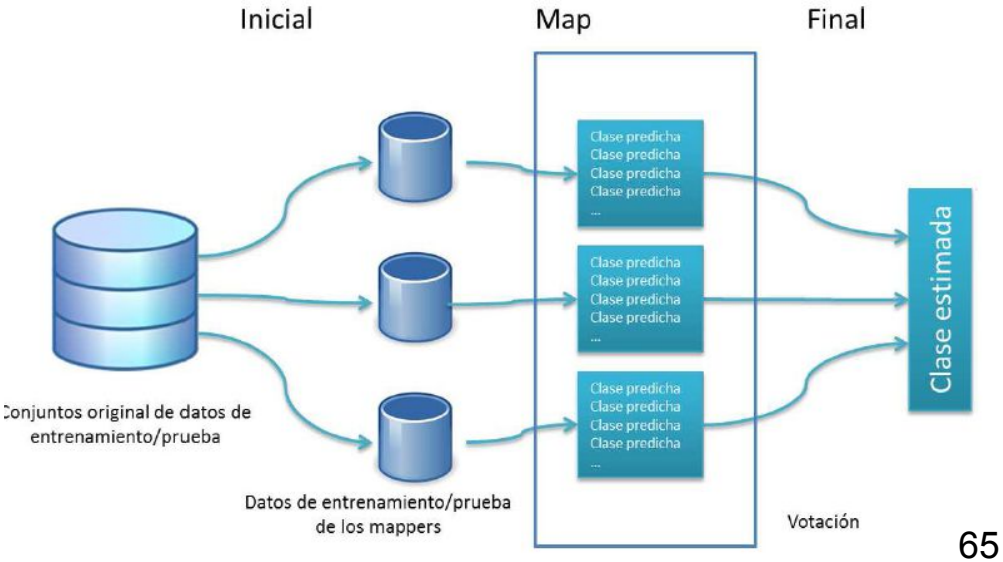
Caso de estudio:
Random Forest para KddCup99

Implementación RF Mahout Partial: Es un algoritmo que genera varios árboles de diferentes partes de los datos (maps).
Dos fases:

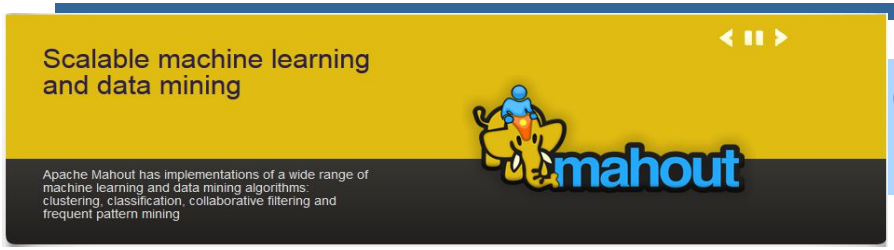
Fase de Construcción



Fase de Clasificación



Mahout. Caso de estudio



Caso de estudio:
Random Forest para KddCup99

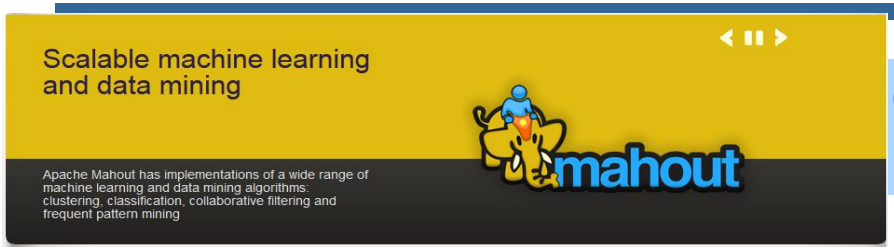
Class	Instance Number
normal	972.781
DOS	3.883.370
PRB	41.102
R2L	1.126
U2R	52

Tiempo en segundos para ejecución secuencial

Datasets	RF		
	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC
DOS_versus_PRB	4825.48	28819.03	NC
DOS_versus_R2L	4454.58	28073.79	NC
DOS_versus_U2R	3848.97	24774.03	NC
normal_versus_PRB	468.75	6011.70	NC
normal_versus_R2L	364.66	4773.09	14703.55
normal_versus_U2R	295.64	4785.66	14635.36

Cluster ATLAS: 16 nodos
-Microprocessors: 2 x Intel E5-2620 (6 cores/12 threads, 2 GHz)
- RAM 64 GB DDR3 ECC 1600MHz
- Mahout version 0.8

Mahout. Caso de estudio



Caso de estudio:
Random Forest para KddCup99

Class	Instance Number
normal	972.781
DOS	3.883.370
PRB	41.102
R2L	1.126
U2R	52

	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC
DOS_versus_PRB	4825.48	28819.03	NC

Tiempo en segundos para Big Data con 20 particiones

Datasets	RF-BigData		
	10%	50%	full
DOS_versus_normal	98	221	236
DOS_versus_PRB	100	186	190
DOS_versus_R2L	97	157	136
DOS_versus_U2R	93	134	122
normal_versus_PRB	94	58	72
normal_versus_R2L	92	39	69
normal_versus_U2R	93	52	64

Cluster ATLAS: 16 nodos
-Spark Random Forest: 43.50 seconds (20 partitions)

Big Data Analytics: Consideraciones

Image Credit: Shutterstock

Without Analytics, Big Data is Just Noise

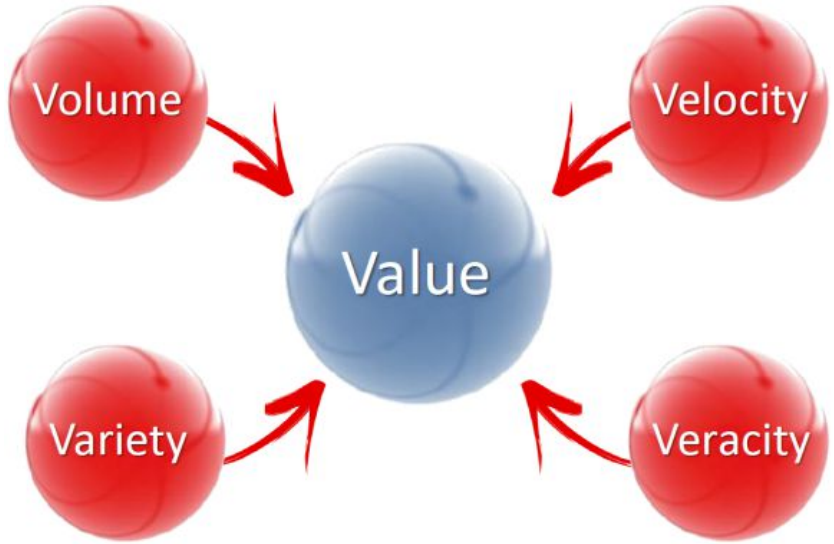
Guest post by Eric Schwartzman, founder and CEO of Comply Socially

http://www.briansolis.com/2013/04/without-analytics-big-data-is-just-noise/

Big Data Analytics → Smart Data

Smart Data

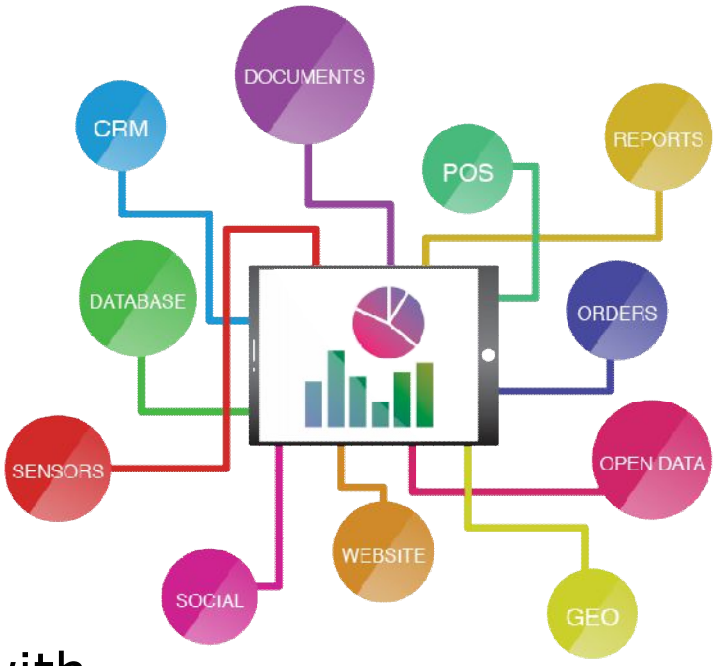
Big data as a concept is defined around five aspects: data volume, data velocity, Data variety and data veracity and data value.



Big Data Analytics → Smart Data

Smart Data

a) While the volume, variety and velocity aspects refer to data generation process and how to capture and store the data,



a) Veracity and value aspects deal with the quality and the usefulness of the data leading to the point.

Big Data Analytics → Smart Data

Smart Data

Smart Data (veracity and value) aims to filter out the noise and hold the valuable data, which can be effectively used by enterprises and governments for planning, operation, monitoring, control, and intelligent decision making.

Big Data Analytics →Smart Data

Smart Data

The key is to explore how Big Data can become Smart Data.

> **"Without Analytics, Big Data is just Noise"**
> ***Eric Schwartzman***

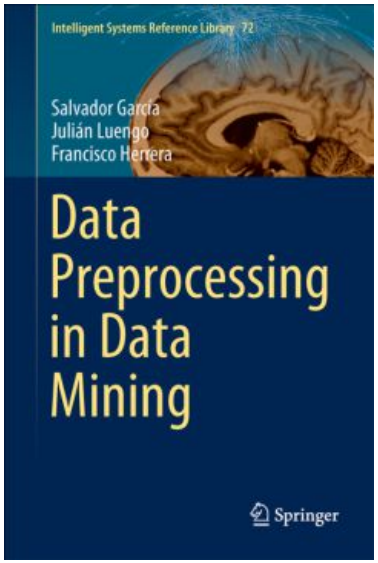
Advanced Big Data modeling and analytics are indispensable for discovering the underlying structure from retrieved data in order to acquire Smart Data.

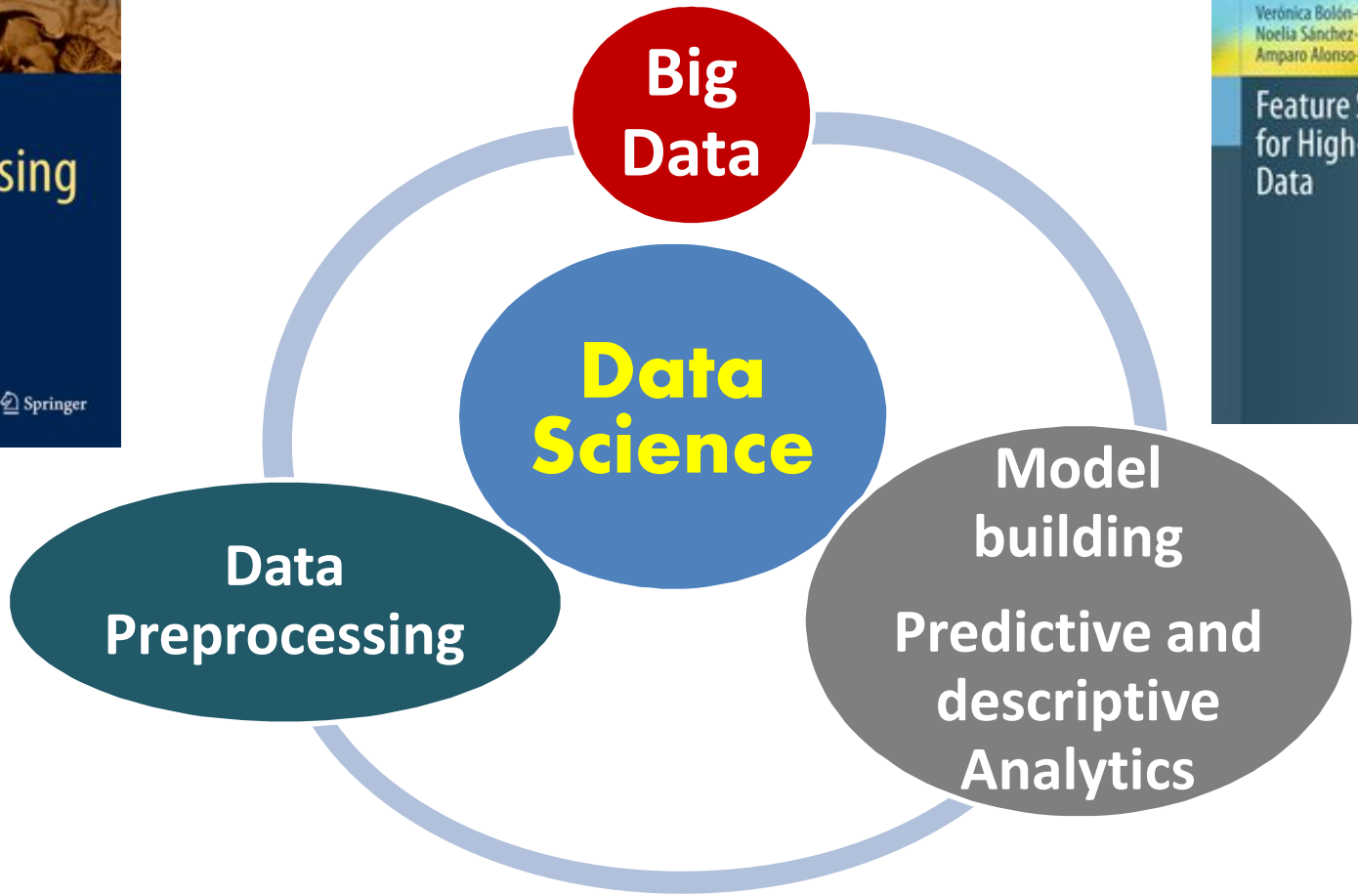
Big Data
+ Analytics
= Smart Data

BIG DATA Preprocessing

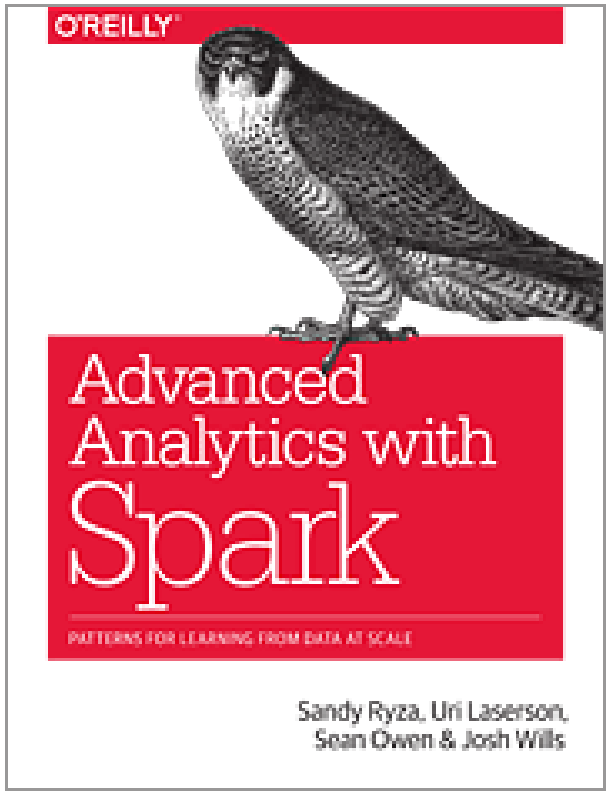
Big Data Analytics:
Big Data Preprocessing

¡Se requieren datos de calidad para diseñar modelos de calidad!.

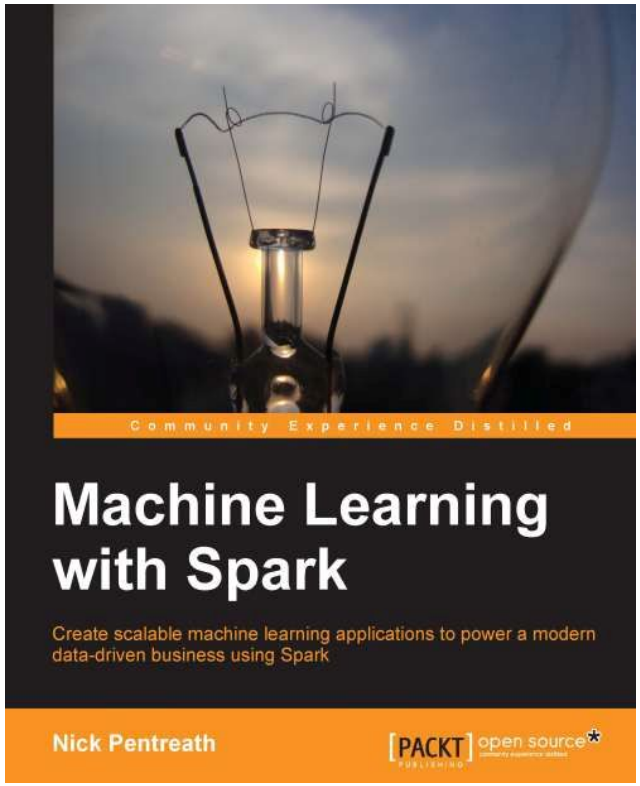




Big Data Analytics: 2 libros

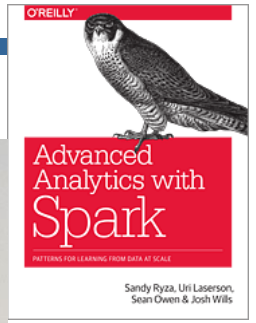


9 cases of study

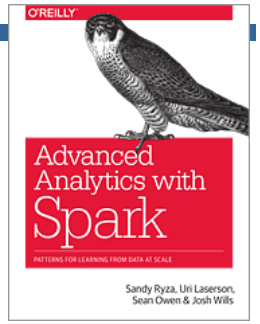


10 chapters giving a quick glance on Machine Learning with Spark

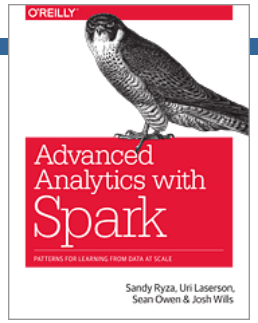
Big Data Analytics: Introducción



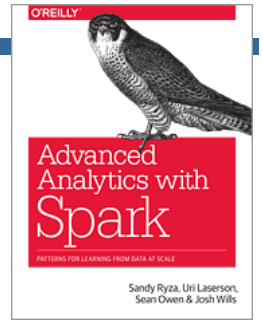
Big Data Analytics: Casos de estudio



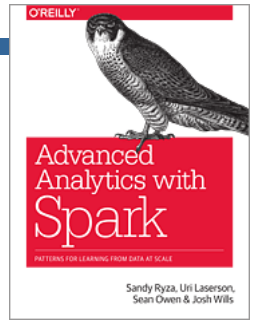
Big Data Analytics: Casos de estudio



Big Data Analytics: Casos de estudio



Big Data Analytics: Casos de estudio



Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

Redes sociales, big data y medidas de salud pública

Studies: Health, Social Media and Big Data

You Are What You Tweet: Analyzing Twitter for Public Health

Discovering Health Topics in Social Media Using Topic Models
Michael J. Paul, Mark Dredze, Johns Hopkins University, Plos One, 2014

Se obtienen 13 grupos coherentes de mensajes correlacionados

- **Gripe estacional (r= 0.689) y alergias (r = 0.810)**
- **Ejercicio y obesidad relacionados con datos geográficos, ..**

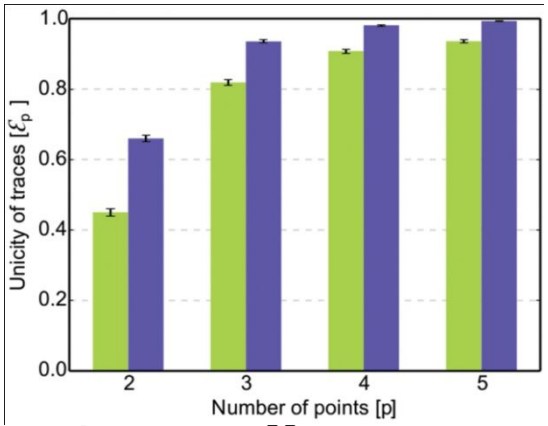
Algunas aplicaciones

Science
AAAS

PRIVACIDAD EN INTERNET »

Cuatro compras con la tarjeta bastan para identificar a cualquier persona

- Los patrones de uso de las tarjetas permiten descubrir la identidad del 90% de una muestra de 1,1 millones de personas anónimas, según demuestra un estudio del MIT

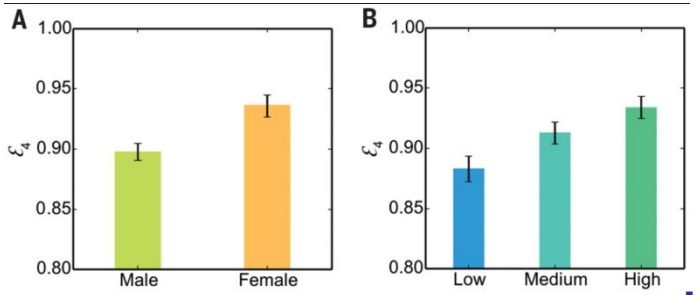


IDENTITY AND PRIVACY

Unique in the shopping mall: On the reidentifiability of credit card metadata

Yves-Alexandre de Montjoye,[1]* Laura Radaelli,[2] Vivek Kumar Singh,[1,3] Alex "Sandy" Pentland[1]

Large-scale data sets of human behavior have the potential to fundamentally transform the way we fight diseases, design cities, or perform research. Metadata, however, contain sensitive information. Understanding the privacy of these data sets is key to their broad use and, ultimately, their impact. We study 3 months of credit card records for 1.1 million people and show that four spatiotemporal points are enough to uniquely reidentify 90% of individuals. We show that knowing the price of a transaction increases the risk of reidentification by 22%, on average. Finally, we show that even data sets that provide coarse information at any or all of the dimensions provide little anonymity and that women are more reidentifiable than men in credit card metadata.



http://www.sciencemag.org/content/347/6221/536

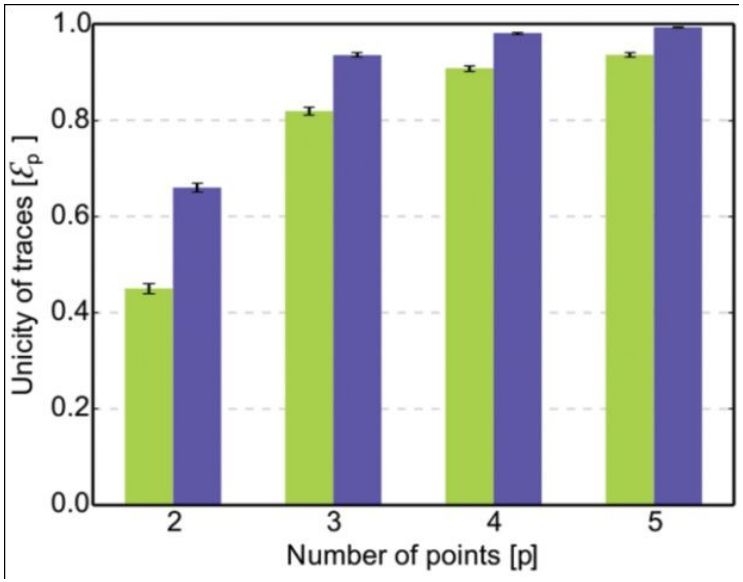
http://elpais.com/elpais/2015/01/29/ciencia/1422520042_066660.html

Banca: Identificación de personas con las compras de tarjetas de crédito

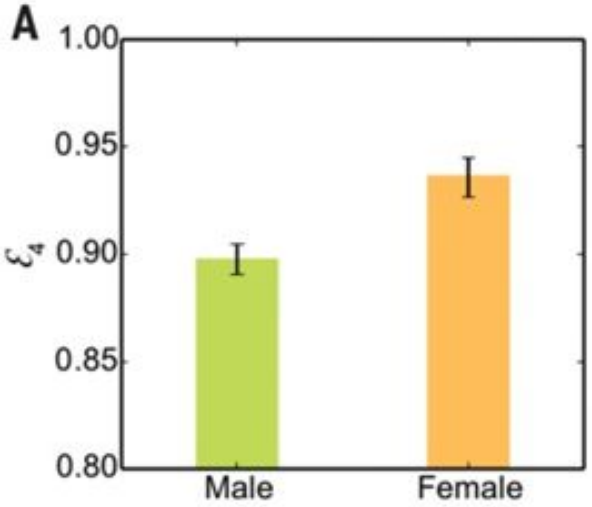
PRIVACIDAD EN INTERNET »

Cuatro compras con la tarjeta bastan para identificar a cualquier persona

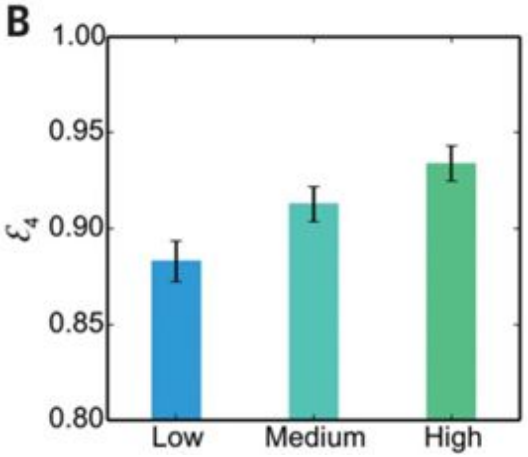
- Los patrones de uso de las tarjetas permiten descubrir la identidad del 90% de una muestra de 1,1 millones de personas anónimas, según demuestra un estudio del MIT



Identificación por el número de compras



Identificación por el género



Identificación por el poder adquisitivo

Análisis de transacciones

Fuente: Big Data. La revolución de los datos masivos. Pag. 77

Análisis de transacciones

Target (cadena de grandes almacenes) que utiliza el análisis de transacciones y asociaciones.

Unos días después el director llamó al padre para disculparse.

Respuesta conciliadora del padre:

"He estado hablando con mi hija –dijo el padre– Resulta que en mi casa han tenido lugar ciertas actividades de las que yo no estaba del todo informado. Mi hija sale de cuentas en agosto. Soy yo el que les debe una disculpa".

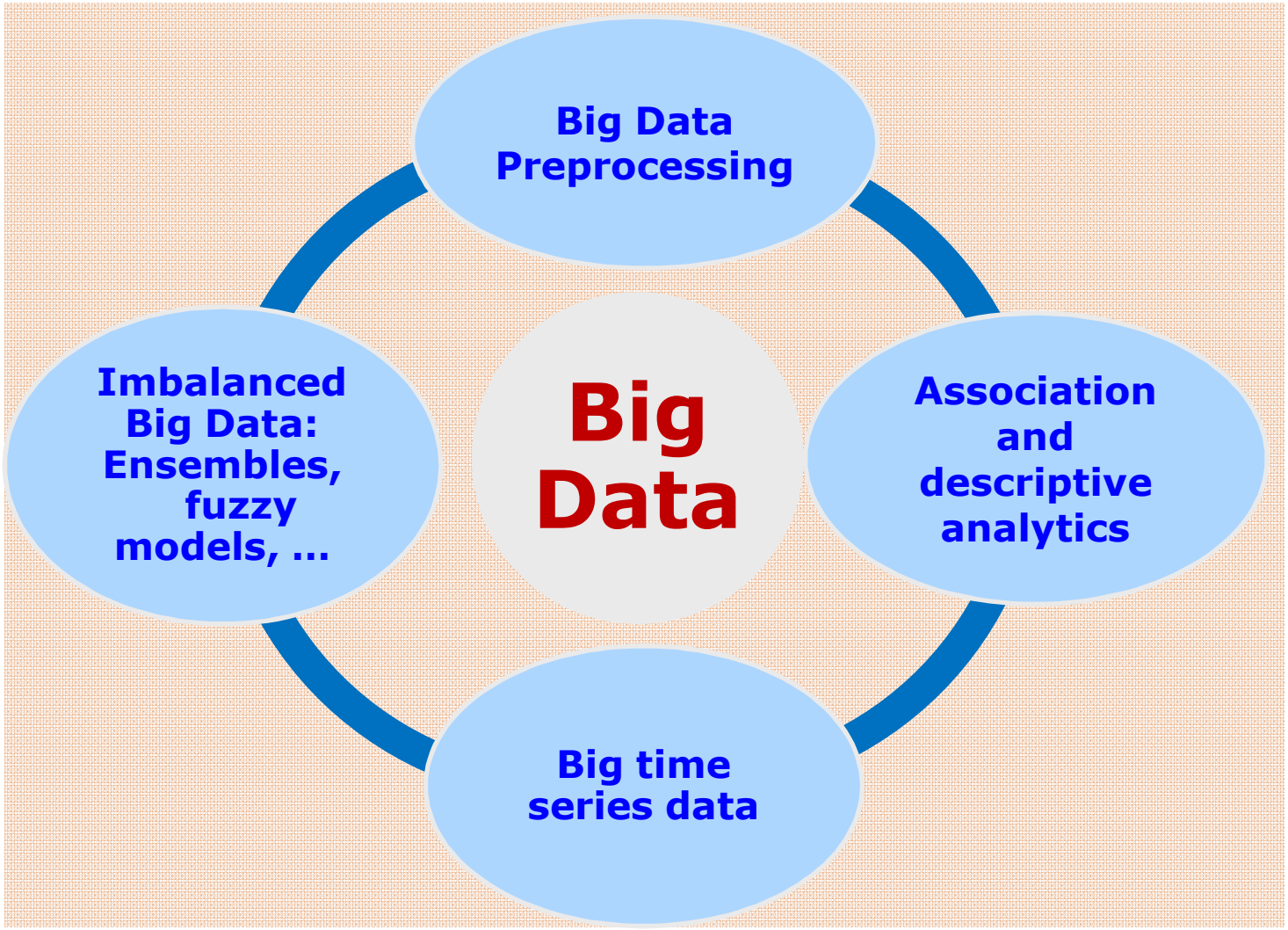
Fuente: Big Data. La revolución de los datos masivos. Pag. 77

Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest
- **Algunas aplicaciones: Salud, Social Media, Identificación**
- Big Data en el grupo de investigación SCI^{2}S
- Comentarios Finales

Big Data at SCI²S - UGR

http://sci2s.ugr.es/BigData



Big Data at SCI²S - UGR

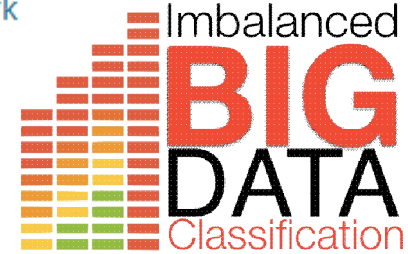
http://sci2s.ugr.es/BigData

SCI²S website

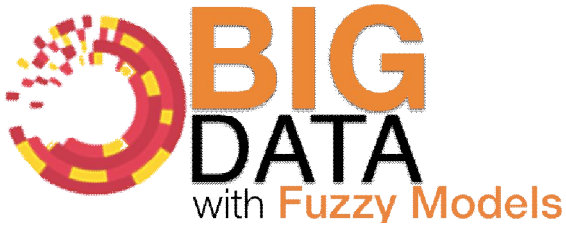
Home » Thematic Sites » Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes

Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes

The web is organized according to the following summary:



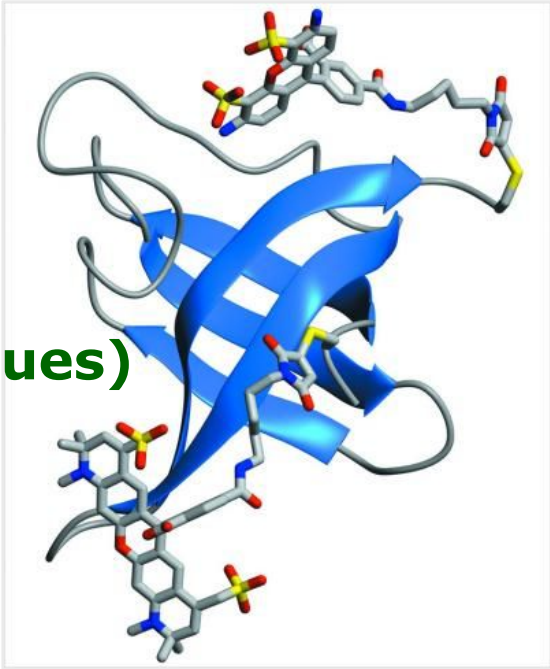
1. Introduction to Big Data
2. Big Data Technologies: Hadoop ecosystem and Spark
3. Big Data preprocessing
4. Imbalanced Big Data classification
5. Big Data classification with fuzzy models
6. Classification Algorithms: k-NN
7. Big Data Applications
8. Dataset Repository
9. Literature review: surveys and overviews
10. Keynote slides
11. Links of interest



This **Website** contains SCI²S research material on algorithms for data preprocessing, computational intelligence and classification with imbalanced datasets in the scenario of Big Data. All information shown here is related to the following SCI²S journal papers and algorithms

Big Data Problem: To visit

ECBDL'14 Big Data Competition 2014: Self-deployment track



Objective: Contact map prediction

Details:

- 32 million instances
- 631 attributes (539 real & 92 nominal values)
- 2 classes
- 98% of negative examples
- About 56.7GB of disk space

Evaluation:

True positive rate · True negative rate

TPR · TNR

http://cruncher.ncl.ac.uk/bdcomp/index.pl?action=data

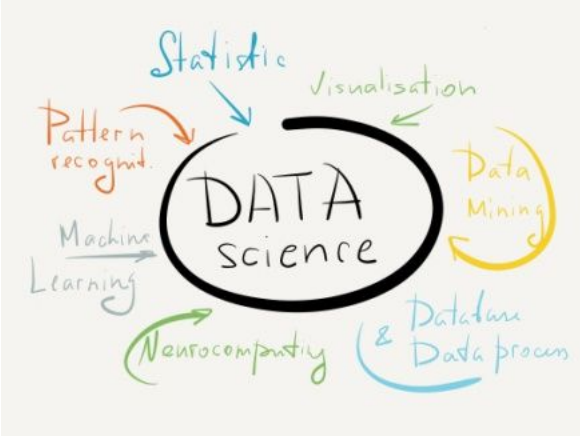
J. Bacardit et al, Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features, Bioinformatics 28 (19) (2012) 2441-2448

Índice

- Big Data. Big Data Science
- ¿Por qué Big Data? Google crea el Modelo de Programación MapReduce
- Tecnologías para Big Data: Ecosistema Hadoop (Hadoop, Spark, …)
- Big Data Analytics: Librerías para Analítica de Datos en Big Data.
- Casos de estudio: Random Forest, Clustering
- Algunas aplicaciones: Salud, Social Media, Identificación
- Big Data en el grupo de investigación SCI²S
- Comentarios Finales

Comentarios Finales

Ciencia de datos: Ámbito del conocimiento que engloba las habilidades asociadas al análisis inteligente de datos, incluyendo Big Data



Científico de datos

Comentarios Finales

Oportunidades en Big Data

Big Data es un área emergente y en expansión. Las posibilidades de desarrollo de algoritmos para nuevos datos, aplicaciones reales ... es un nicho de investigación y desarrollo en los próximos años.

Comentarios Finales

http://elpais.com/elpais/2015/03/26/buenavida/1427382655_646798.html

¿Qué es eso del 'big data'?

- Lo mencionan en conferencias, charlas y facultades. Aclaramos el concepto de moda. O lo que es lo mismo: lo que todas las empresas quieren saber de usted

EVA VAN DEN BERG | 31 MAR 2015 - 12:28 CEST

Tiene continua repercusión en la prensa

Comentarios Finales

Big Data: Gran Impacto en la Sociedad y presencia en los medios de comun

ELMUNDO.es
Líder mundial en español | Miércoles 04/09/2013. Actualizado 16:27h.

INTERNET | Campus Party Europa 2013

'Es la década de los ahí vendrá la revolu

Alex 'Sandy' Pentland, durante su ponencia. | M. Sáinz

http://www.elmundo.es/elmundo/

EL PAÍS
ECONOMÍA
ECONOMÍA EMPRESAS MERCADOS BOLSA

El maná de los datos

- La conversión de datos en información útil para la millones de dólares en 2015. La herramienta 'big

SUSANA BLÁZQUEZ | Madrid | 29 SEP 2013 - 01:00 C

Archivado en: Citigroup Cap Gemini Sogeti SAP IBM Telefónica Aplicaciones informáticas Tecnología

EL PAÍS
ECONOMÍA
ECONOMÍA EMPRESAS MERCADOS BOLSA MIS AHORROS VIVIENDA TECNOLOGÍA OF

EMPRENDEDORES »

El Big Data echa una mano al campo

- Una empresa española recoge miles de datos para predecir las cosechas

MARÍA FERNÁNDEZ | 30 NOV 2014 - 00:00 CET

Archivado en: Bases datos Emprendedores Aplicaciones informáticas Empresas Programas informáticos Economía Informática Industria

Comentarios Finales

- La paralelización de los algoritmos de aprendizaje automático junto al particionamiento de datos pueden proporcionar algoritmos de calidad con MapReduce.
  - Paticionando datos y aplicando el algoritmo a cada parte.
  - Centrando la atención en la fase de combinacion (**reduce**). La combinación de modelos es un reto en el diseño de cada algoritmo.
- Data Mining, Machine learning and data preprocessing: Inmensa colección de algoritmos frente a los pocos algoritmos en big data analytics.

Comentarios Finales

Data Mining, Machine learning and data preprocessing: Inmensa colección de algoritmos

Big Data Analytics

Big Data: Un pequeño conjunto de algoritmos

Big Data Preprocessing:

Unos pocos métodos de preprocesamiento.

Comentarios Finales

- Para el diseño y/o adaptación de cualquier algoritmo es necesario diseñar de forma adecuada una fase de fusión de información por cuanto siempre será necesario utilizar funciones Map y Reduce cuando la base de datos sea muy grande.
- Igualmente los procedimientos iterativos requieren de un diseño adecuado para optimizar la eficiencia.
- Todavía se está en una fase muy temprana de diseño de algoritmos de aprendizaje automático para big data.
- El preprocesamiento de datos es esencial para mejorar el comportamiento de los algoritmos de aprendizaje. El diseño de estos algoritmos para big data está en una fase muy incipiente.

Comentarios Finales

Big data and analytics:
Un gran reto que ofrece múltiples oportunidades

- **Pequeño conjunto de algoritmos**
  Es necesario rediseñar nuevos algoritmos.
- **Modelo de Computación**
  - **Precisión y aproximación**
  - **Requiere "eficiencia" en los algoritmos.**

- **Datos de calidad para modelos de calidad en big data**
  Modelos/Decisiones de calidad están basados en datos de calidad.
  - **Preprocesamiento en Big Data**
  - **Análisis del ruido en datos**
    Métodos automáticos de limpieza
  - **Procesamiento de valores perdidos**
  - **Big Data Reduction**

Comentarios Finales

Una demanda creciente de profesionales en "Big Data" y "Ciencia de Datos"

Oportunidades en Big Data (en España)

http://www.revistacloudcomputing.com/2013/10/espana-necesitara-60-000-profesionales-de-big-data-hasta-2015/?goback=.gde_4377072_member_5811011886832984067#!

España necesitará 60.000 profesionales de Big Data hasta 2015

22 octubre, 2013 Eventos 18

España necesitará 60.000 profesionales de Big Data hasta 2015

"España va a necesitar alrededor de sesenta mil profesionales del Big Data de aquí a 2015", así lo ha asegurado Francisco Javier Antón, Subdirector General de Tecnologías del Ministerio de Educación, Cultura y Deportes en una mesa redonda sobre beneficio y aplicación de Big Data en pymes, moderada por Daniel Tapias de Sigma Technologies, celebrada durante el 4º Congreso Nacional de CENTAC de Toledo.

"Existe una demanda mundial para formar a 4,4 millones de profesionales de la gestión Big Data desde ingenieros, gestores y científicos de datos", comenta Antón. Sin embargo, "las empresas todavía no ven en el Big Data un modelo de negocio", lamenta. "Solo se extrae un 1% de los datos disponibles en la red", añade. "Hace falta formación y concienciación.

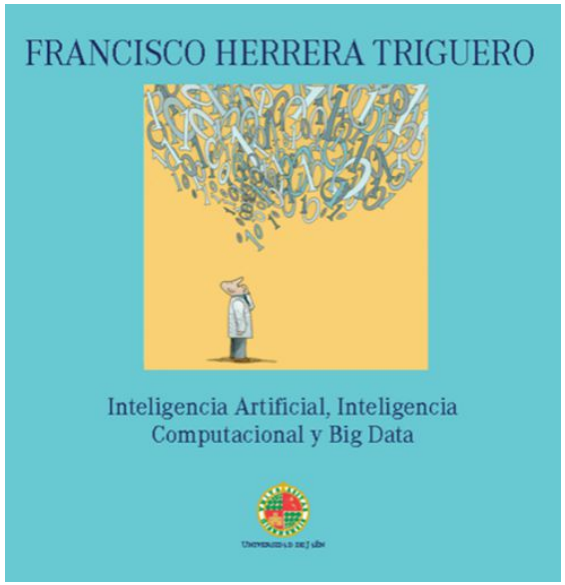
Comentarios Finales

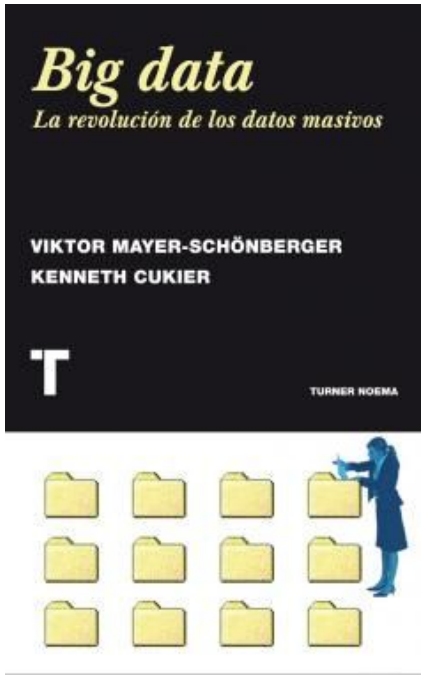
Las grandes colecciones de datos nos van a conducir a nuevas innovaciones (en la industria, ciencia, sociedad)

Ben Chams - Fotolia

Comentarios Finales

2 Lecturas rápidas:





Capítulo 3.

http://issuu.com/secacult_uja/docs/libro_francisco_herrera.indd

A. Fernandez, S. Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, **Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks.** *WIREs Data Mining and Knowledge Discovery 4:5 (2014) 380-409*

BIG DATA

Big Data