Técnicas de *Clasificación:* **EJEMPLOS**

- **Paradigma Inductivo**. Árboles de decisión:
  - ID3,
  - CART,
  - C4.5,
  - See5,
  - **Random Forest** (de moda en Big Data), Leo Breiman 2001,
  - Etc.

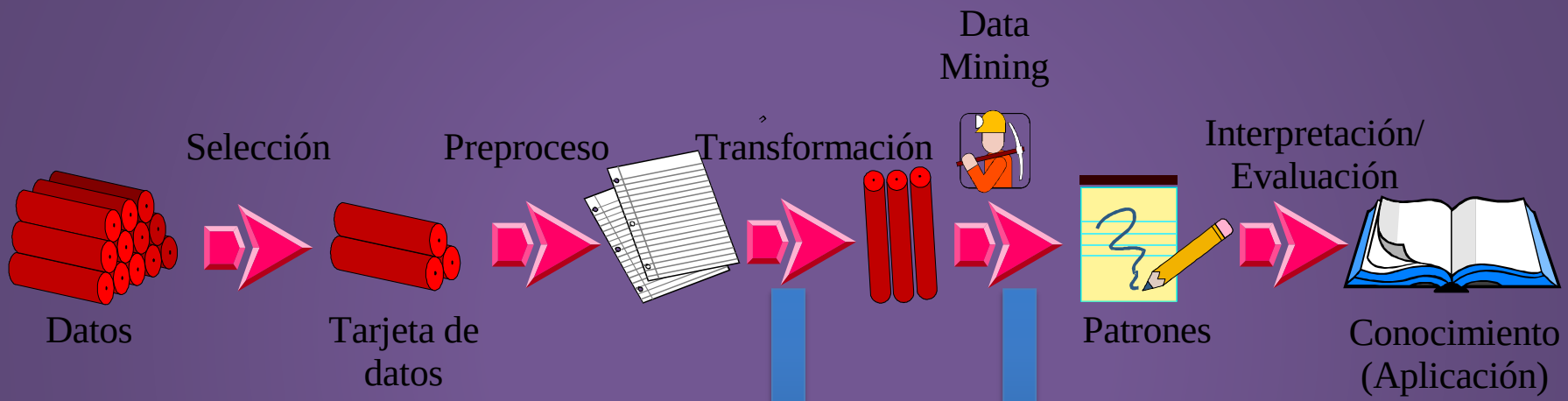
Técnicas de *Clasificación:* **EJEMPLOS**

- **Paradigma Conexionista**. Redes Neuronales Artificiales:
  - Perceptrón Multicapa (con backpropagation),
  - Convolucionales,
  - Neocognitrones,
  - Redes de Hopfield,
  - Redes recurrentes,
  - Adaline,
  - **Deep Learning** (de moda en Big Data),
  - Etc.

Técnicas de *Clasificación:* **EJEMPLOS**

- **Modelos estadísticos y probabilistas**.
  - Redes Bayesianas,
  - Naive-Bayes,
  - Máquinas de Soporte Vectorial (SVM),
  - Metaheurísticas,
  - Etc.

La importancia del KDD



Consisten habitualmente en convertir un proceso de ‘Clustering’ en uno de ‘clasificación’.

Adecuación de los métodos a los problemas.

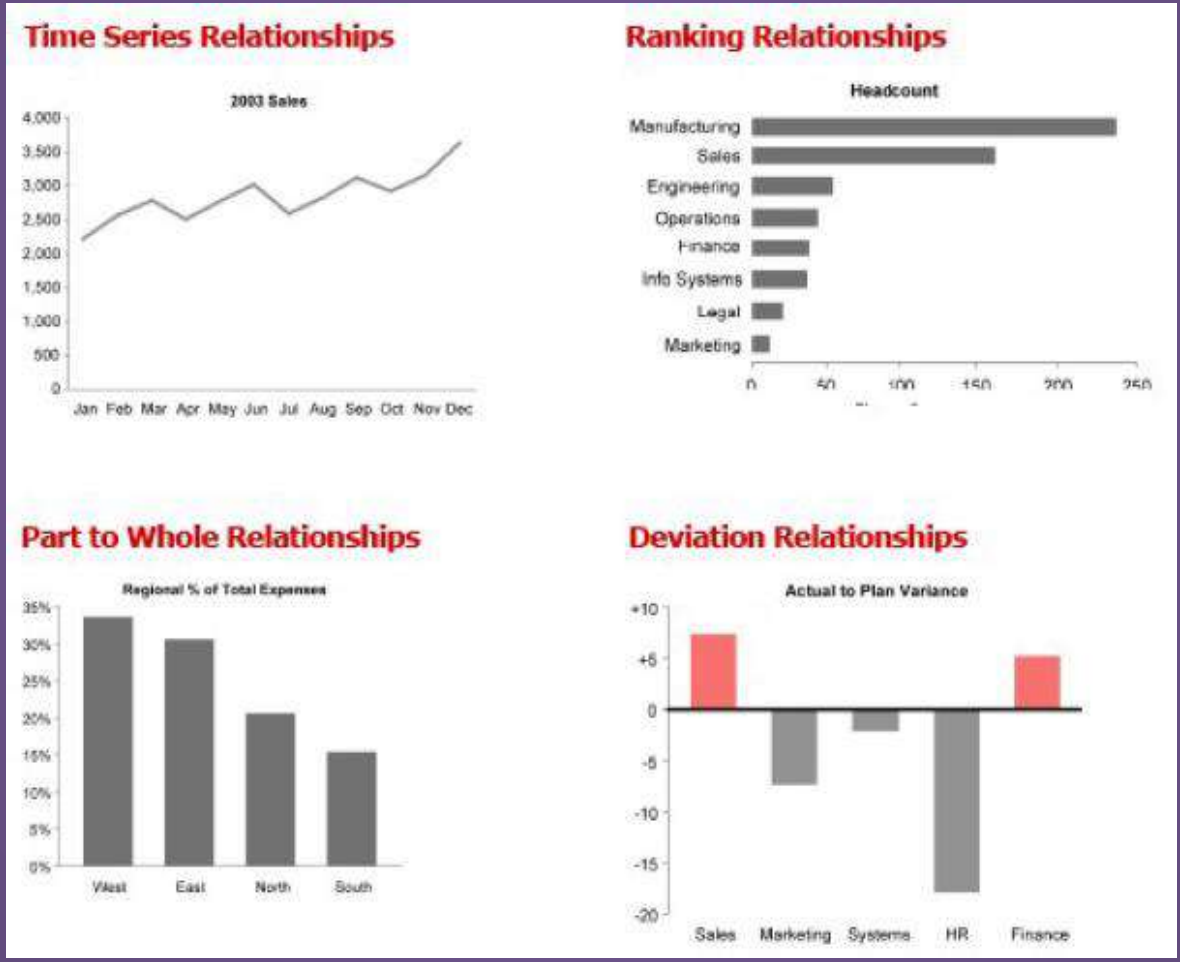
Análisis Descriptivo

- Resumen claro y fácil de entender de una colección de datos.
- Fundamento y concepto más básico de todas las estadísticas.
- Visualización de datos para entender el pasado y el presente.

Análisis Descriptivo

- Se describen los datos con tablas o gráficos.
- Descripciones numéricas de la variabilidad y la posición.
- Modelización descriptiva.

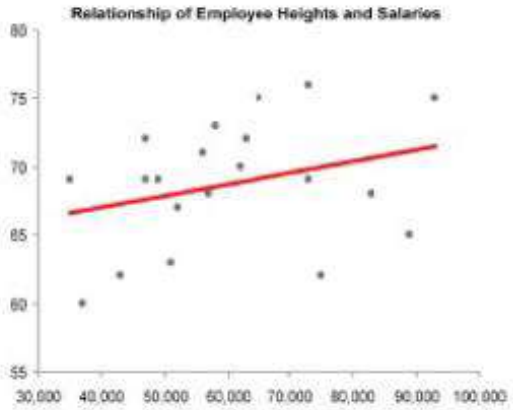
Análisis Descriptivo



Análisis Descriptivo

Distribution Relationships

Correlation Relationships



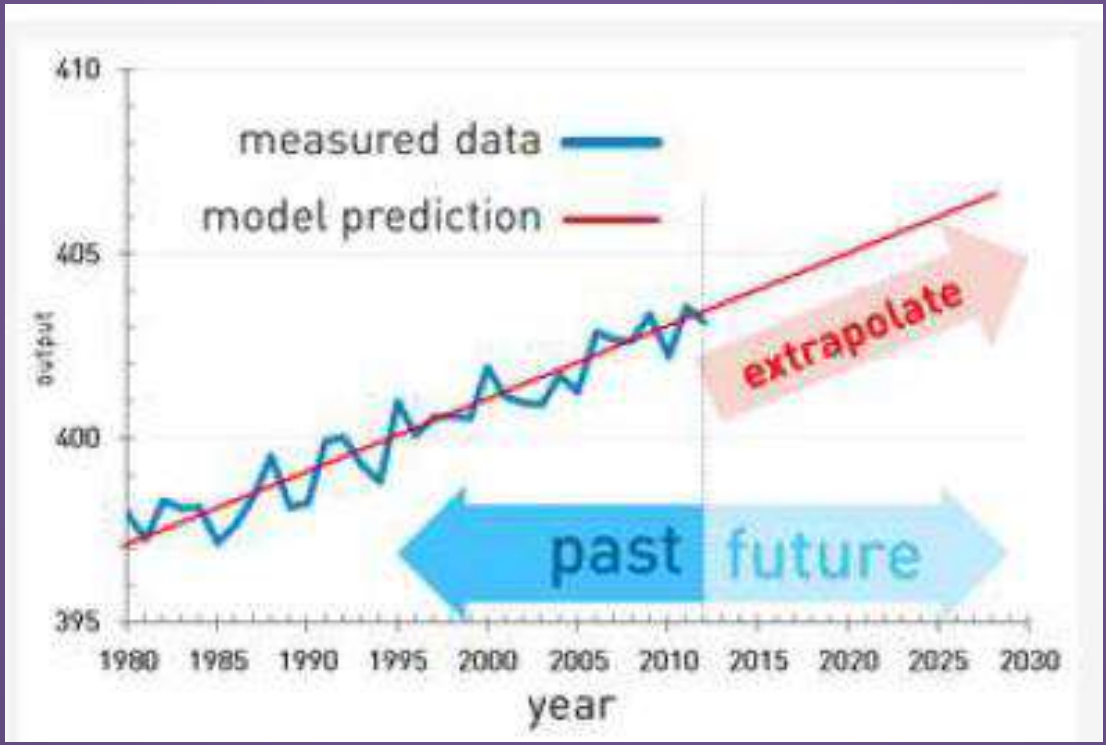
Nominal Comparison Relationships

Análisis Predictivo

- **Extrapolación de funciones** (Tendencia para el futuro, pero no hay capacidad de pronóstico –hechos/cambios puntuales-).
- **Correlaciones entre variables** (Demasiado evidente, no suele funcionar de forma muy fina).
- Encontrar **‘patrones’** en los datos que puedan ser aplicados a situaciones futuras (KDD y Data Mining).
- Métodos de **CLUSTERING** Y **CLASIFICACIÓN**.

Análisis Predictivo

- **Extrapolación de funciones** (Por ejemplo Estimaciones o Líneas de Tendencia).



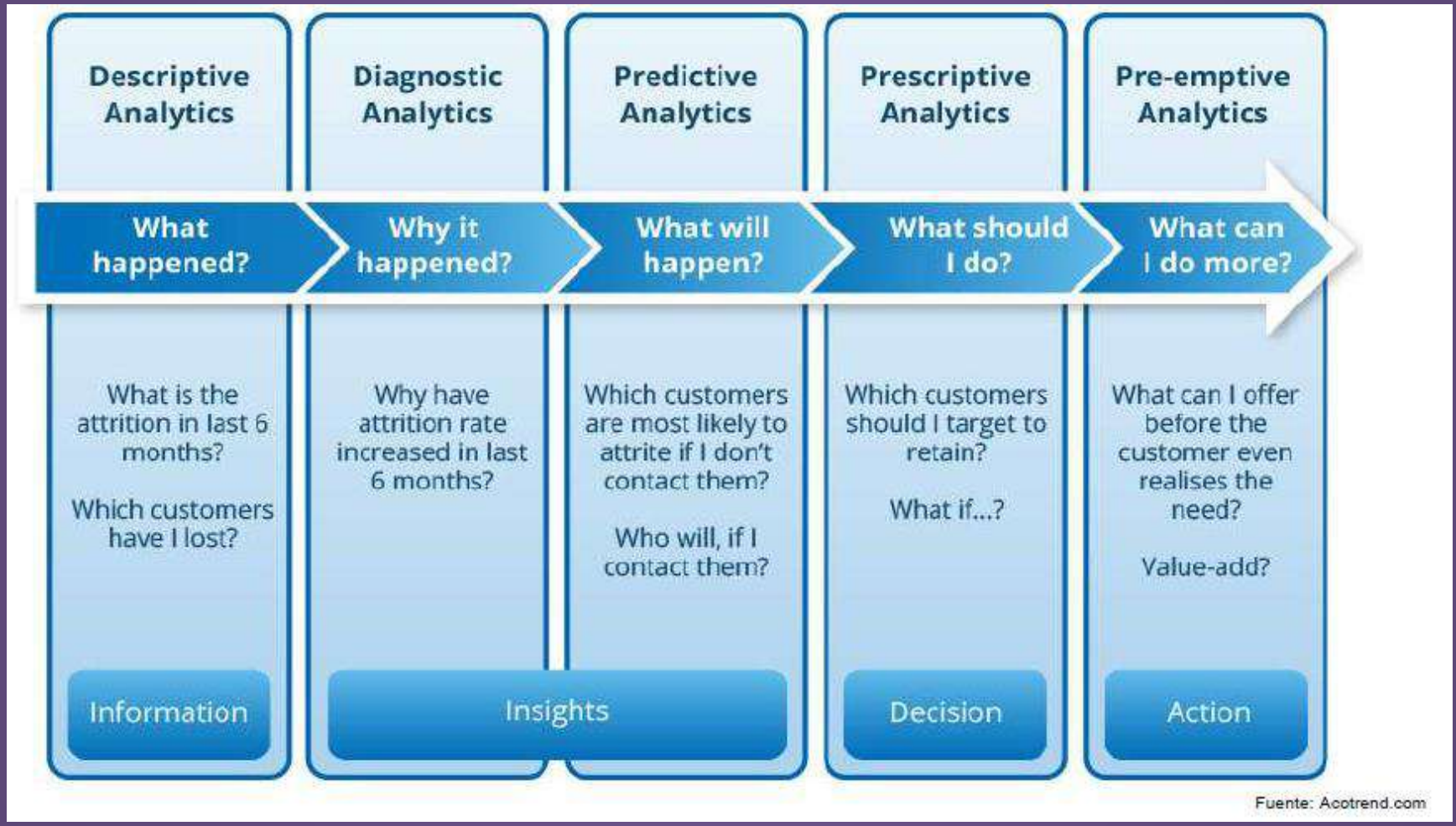
Análisis Prescriptivo

- El análisis predictivo se centra en **un escenario** futuro.
- El prescriptivo se centra en **múltiples alternativas**.
- Por lo tanto, un modelo prescriptivo puede ser considerado como **una combinación de modelos predictivos** (uno por cada posible escenario), que **se ejecutan en paralelo**.
- El objetivo es encontrar la mejor opción posible: **OPTIMIZACIÓN**.

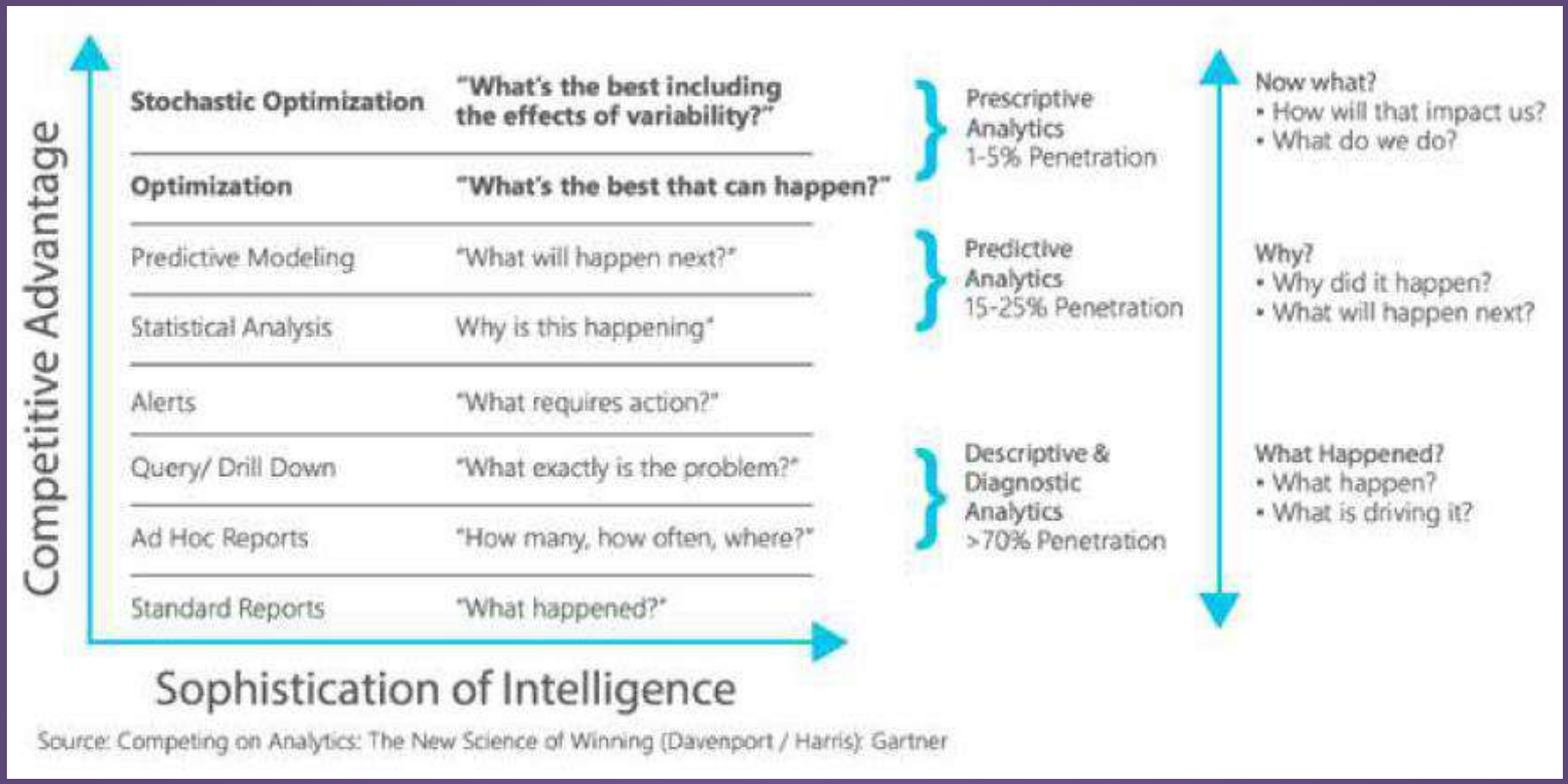
Análisis Prescriptivo

- Técnicas:
  - Técnicas de Investigación Operativa,
  - Algoritmos Genéticos,
  - Técnicas estocásticas,
  - Metaheurísticas,
  - Etc.

Análisis Prescriptivo



Análisis Prescriptivo



Source: Competing on Analytics: The New Science of Winning (Davenport / Harris); Gartner

Herramientas actuales (en entornos Big Data) para implementar estas soluciones.

MapReduce

- Una sola máquina no puede gestionar **una gran cantidad** de datos de manera **eficiente**.
- El problema es la **escalalabilidad** para trabajar con grandes volúmenes de datos.
- Una posible solución: Adoptar una estrategia **Divide y Vencerás**.
- Además los volúmenes de datos **crecen** constante y vertiginosamente y las **restricciones de tiempo** se mantienen.
- Se solía aumentar el número de nodos de proceso (**incremento de costes, espacio**…).

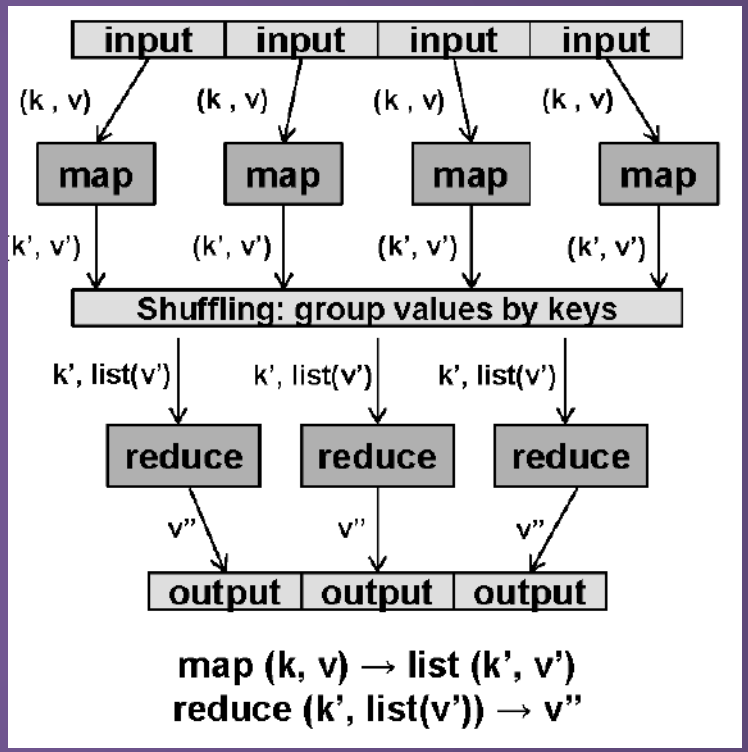
MapReduce

- **Google 2004**: Paradigma **MapReduce**:
  - En 2004 capaz de procesar 20Pb de datos al día.
  - Modelo **simple** y **elegante**, con capacidad de extenderse a muchas aplicaciones.
  - Modelo de **programación en paralelo**.
- Se hizo popular con **Hadoop** (proyecto de código abierto):
  - Entre otros, lo usan **Yahoo!**, **Facebook**, **Amazon**…
- Es el entorno más popular para Big Data.

J. Dean, S. Ghemawat: "MapReduce: Simplified data processing on large clusters", Communications of the ACM 51, 2008.

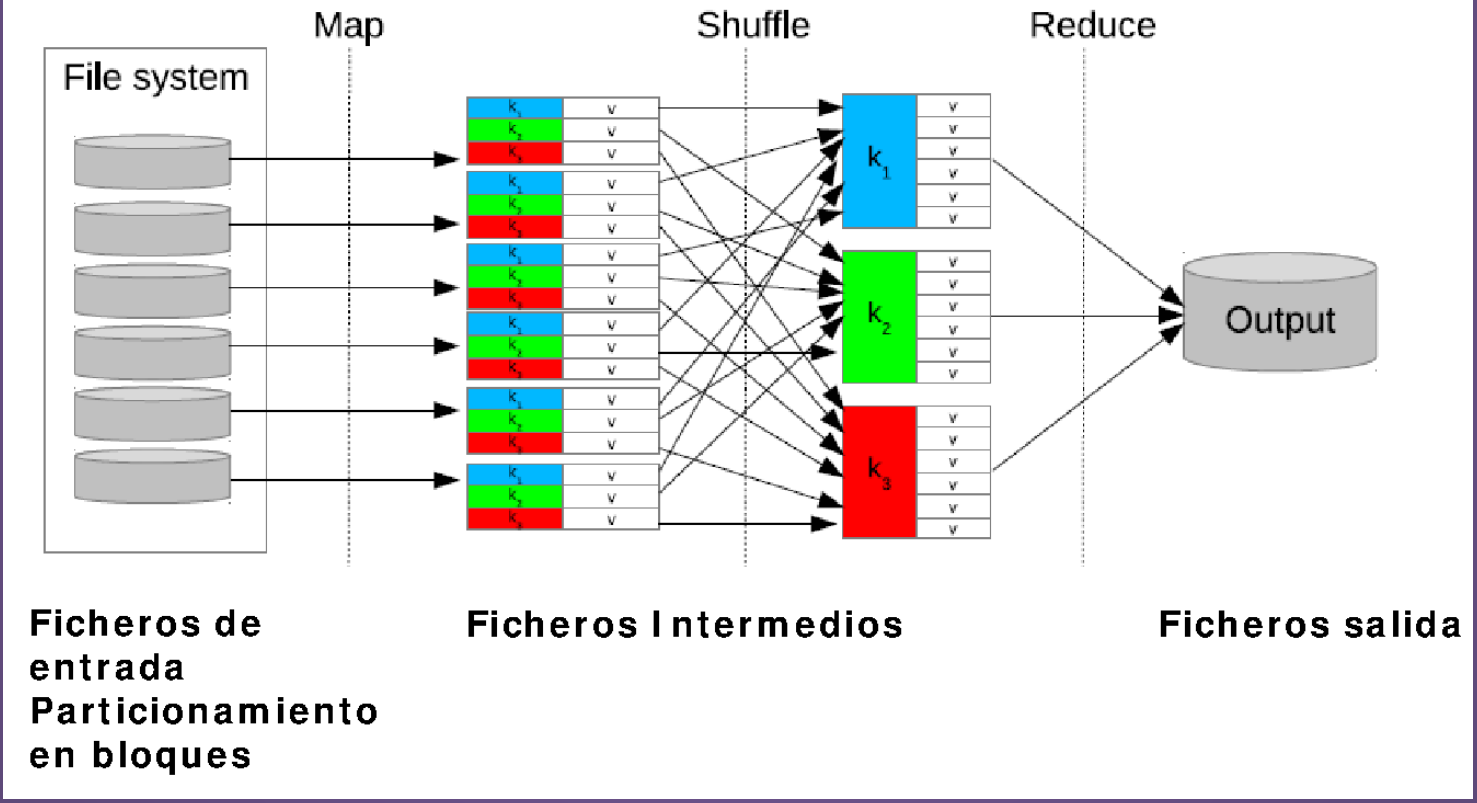
MapReduce

- Tiene una estructura intermedia **Valor-Llave**.
- Se llevan a cabo 2 operaciones:
  - Función **Map**: Procesa bloques de información.
  - Función **Reduce**: Fusiona los resultados previos de acuerdo a su llave.
  - Hay una etapa intermedia de agrupamiento por llave (**Shuffling**).



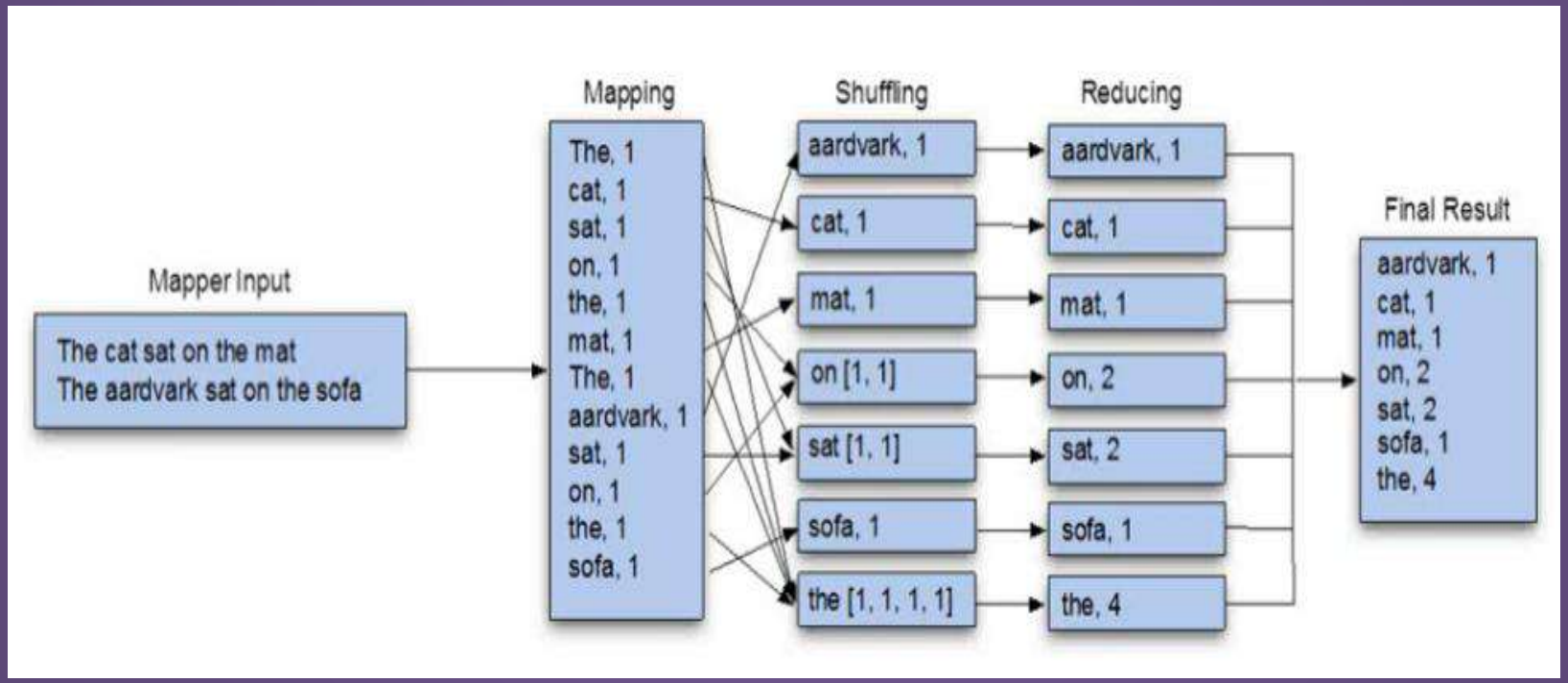
MapReduce

- Cómo trabaja Map Reduce, transparente para el programador:



MapReduce

- Cómo trabaja Map Reduce, transparente para el programador (ejemplo):

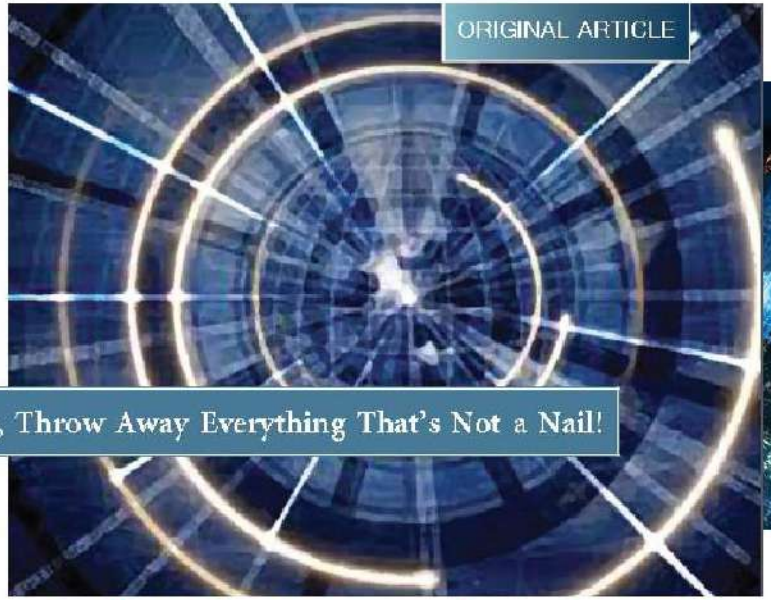


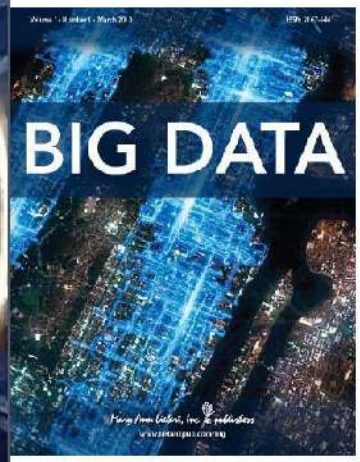
MapReduce

- Por lo tanto…
  - El modelo MapReduce tiene **ventajas frente a los modelos distribuidos clásicos** porque oculta la complejidad de la distribución y la tolerancia a fallos en su paralelismo.
  - Es **escalable**: se reducen los problemas de hardware.
  - Es **más barato**: ahorro en costes de hardware, programación, administración…
  - Cuando funciona puede **ahorrar mucho tiempo**.
- Pero…
  - **No** es **adecuado** para **todos** los **problemas**.

MapReduce

MAPREDUCE IS GOOD ENOUGH?





If All You Have is a Hammer, Throw Away Everything That's Not a Nail!

Jimmy Lin
The iSchool, University of Maryland
College Park, Maryland

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

Iterative Graph Algorithms
Gradient Descent
Expectation Maximization

MapReduce

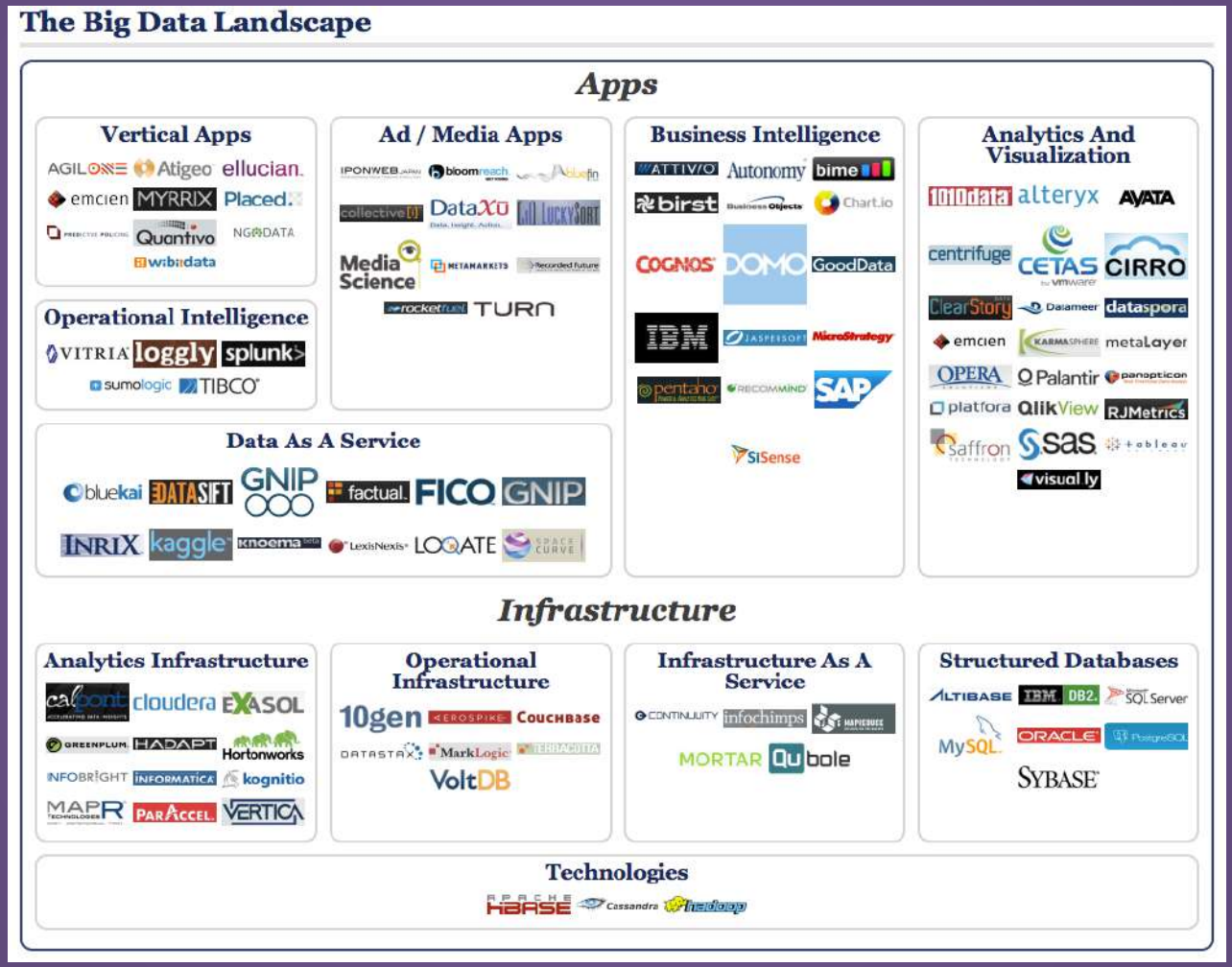
- Hay muchas **limitaciones** para los **algoritmos de grafos iterativos**.
- Por ejemplo, en PageRank, cada iteración es una ejecución completa de MapReduce.
- Por ello, se han propuesto una serie de **extensiones** para mejorar el cálculo iterativo.
- Por ejemplo **Pregel** (Google) http://www.michaelnielsen.org/ddi/pregel

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

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
2013
essir
granada.spain
Enrique Alfonseca
Google Research Zurich

Software existente



Hadoop

- Es una **implementación en código abierto** del paradigma computacional **MapReduce**.
- Hadoop Distributed File System (**HDFS**) es un sistema de archivos escrito en Java, distribuido, escalable y portátil.
- Creado por **Doug Cutting** (Apache Software Foundation) en 2010.

http://hadoop.apache.org

Hadoop

- Hadoop es una nueva forma de almacenar y analizar datos para las empresas:
  - Inspirado en ficheros Google GFS y arquitectura MapReduce.
  - Escalabilidad horizontal, cuando sea necesario.
  - Data Accesss Bottleneck (Localidad en procesamiento de los datos).
  - Disk Performance Bottleneck. (uso de múltiples discos en paralelo) Ej. 250/node cluster with 4 disks per node = 1000 disks
  - Uso de un lenguaje popular (MapReduce escrito en APIs de Java).
  - Fault Tolerance: Equates about one failure per day in a 2,000 node cluster.

http://hadoop.apache.org

Hadoop

¿Cómo acceder a una plataforma Hadoop?

- Plataformas cloud con instalación de Hadoop:
  - Amazon Elastic Compute Cloud.

http://aws.amazon.com/es/ec2

  - Windows Azure.

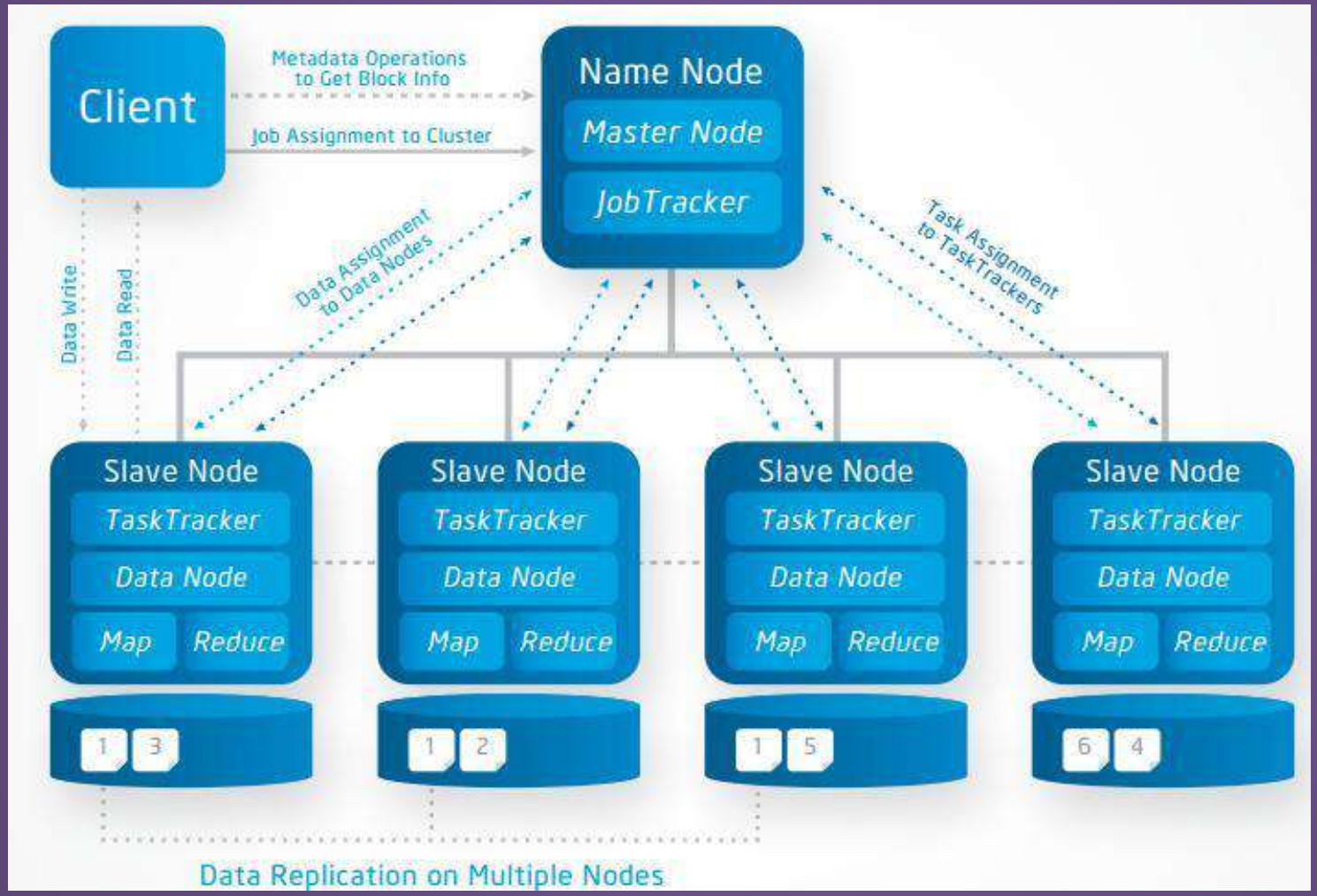
http://www.windowsazure.com

  - Instalación en un cluster: Cloudera para Hadoop.

http://www.cloudera.com/why-cloudera.html

Ecosistema Hadoop

Arquitectura Hadoop



Ecosistema Hadoop

Ecosistema Hadoop

- Apache FLUME:
  - “Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows.”
- Apache Hbase:
  - “Apache HBase is the Hadoop database, a distributed, scalable, big data store. Use Apache HBase™ when you need random, realtime read/write access to your Big Data.”

Ecosistema Hadoop

- Apache Oozie:
  - “Oozie is a workflow scheduler system to manage Apache Hadoop jobs.”
- Apache Lucene:
  - “Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java.”
- Apache Pig:
  - “Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.” (Similar a SQL, genera MR).

Ecosistema Hadoop

- IBM BigSheets:
  - “The power of Apache Hadoop with an accessible, easy-to-use, web based front end: BigSheets makes Do It Yourself Analytics into a reality, BigSheets can help your line of business professionals dive deep into your data.” (Como Excel a lo grande, genera MR).

Spark

- Apache Spark:
  - Permite trabajos paralelizados **totalmente en memoria**, lo que reduce mucho los tiempos de proceso.
  - Sobre todo si se trata de **procesos iterativos**.
  - Si algunos datos no caben en memoria, Spark sigue trabajando y usa el disco duro para volcar aquellos datos que no se necesitan en ese momento (**Hadoop "commodity hardware"**).
  - Esquema de computación **más flexible** que MapReduce: Permite **flujos acíclicos** y **algoritmos iterativos**.
  - Ofrece una API para Java, Python y Scala.

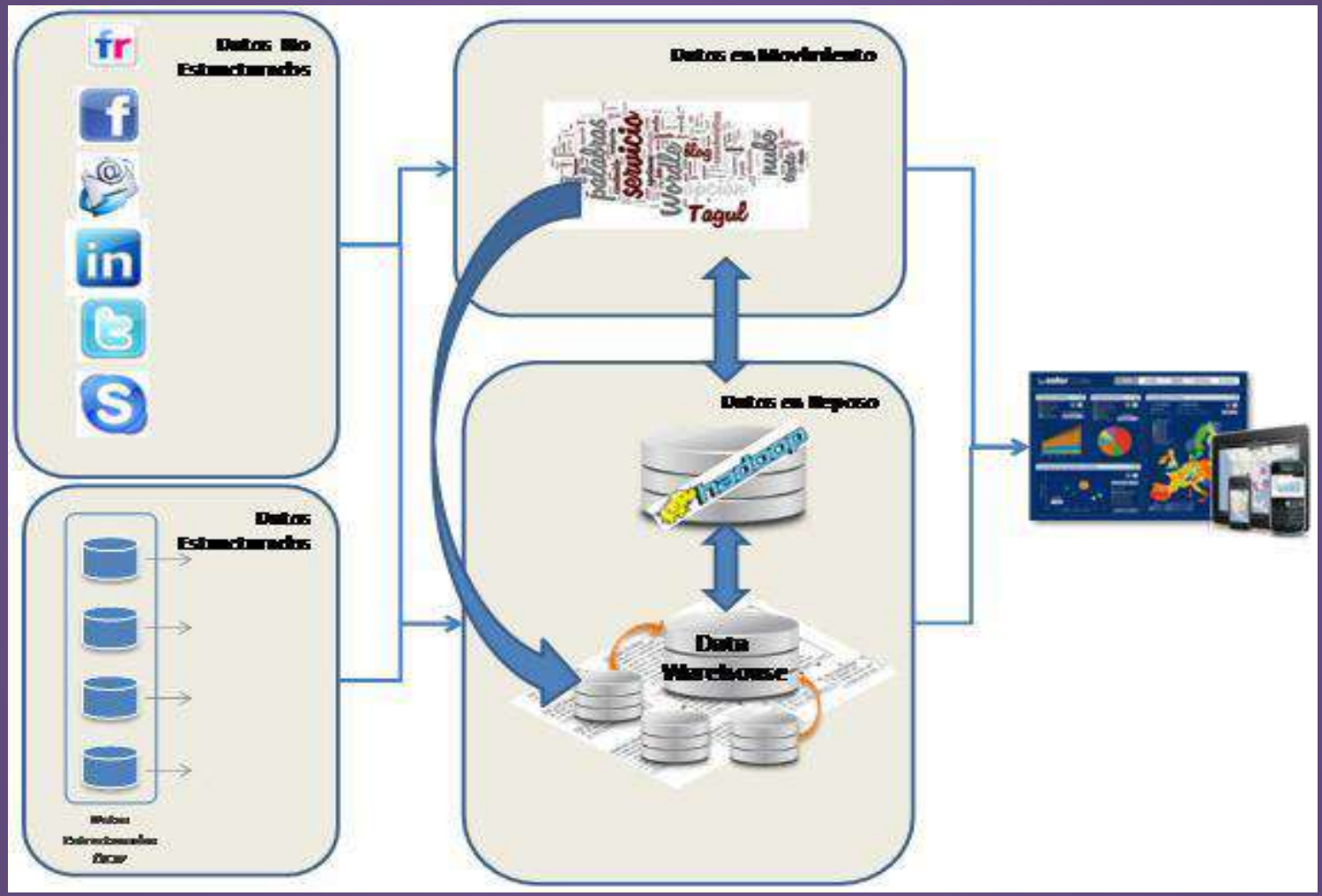
Spark

- Apache Spark:
  - "Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk."
  - "Combine SQL, streaming, and complex analytics."
  - "Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3."
  - Funcional, fácil de programar.

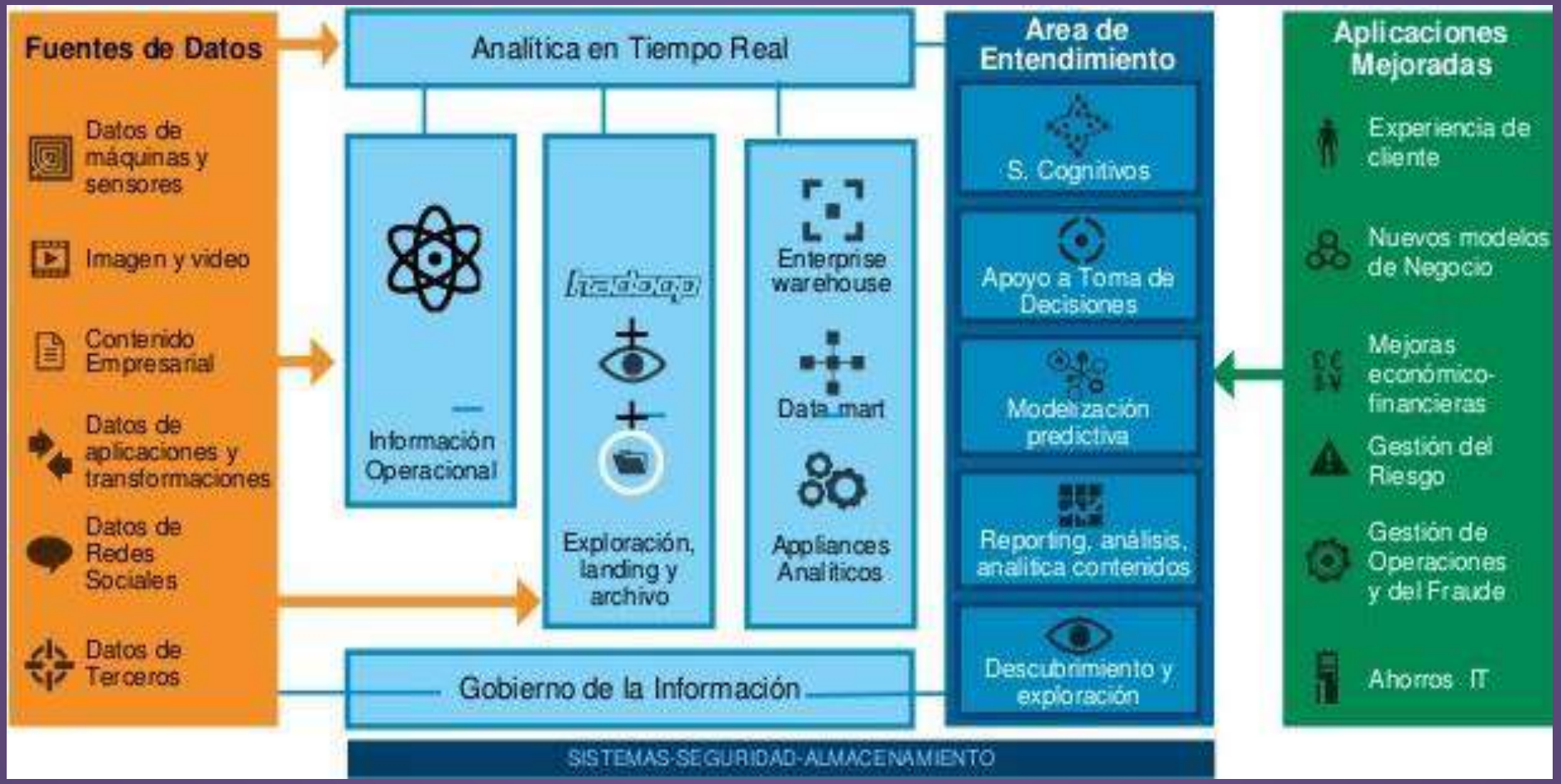
Spark

- Apache Spark Shells:
  - Clase RDD: “A Resilient Distributed Dataset (RDD), the basic abstraction in Spark. Represents an immutable, partitioned collection of elements that can be operated on in parallel.”
  - Apache AMBARI: “The Apache Ambari project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.”

Arquitectura Big Data – una visión



Arquitectura Big Data – otra visión



Reflexión

Una reflexión tecnológica (si resulta adecuada…):

- Algoritmos basados en el descenso del gradiente.
- Algoritmos dependientes y no dependientes de la iteración anterior, modelos aleatorios…
- Procesos con flujos acíclicos de procesamiento de datos.

http://tez.apache.org/

Reflexión

Una reflexión tecnológica (si resulta adecuada…):

- Leo Breiman…
  - Random Forest, Bagging (Bootstrap aggregation) y
  - Boosted Regresion Trees (secuencial, iterativo).
- Deep Learning…
  - Cybenko, backpropagation…

Librerías para Big Data

Spark Libraries

https:/ / spark.apache.org/ releases/ spark-release-1-3-0.html

Spark SQL

In this release Spark SQL graduates from an alpha project, providing backwards compatibility guarantees for the HiveQL dialect and stable programmatic API's. Spark SQL adds support for writing tables in the data sources API. A new JDBC data source allows importing and exporting from MySQL, Postgres, and other RDBMS systems. A variety of small changes have expanded the coverage of HiveQL in Spark SQL. Spark SQL also adds support schema evolution with the ability to merging compatible schemas in Parquet.

Spark ML/MLlib

In this release Spark MLlib introduces several new algorithms: latent Dirichlet allocation (LDA) for topic modeling, multinomial logistic regression for multiclass classification, Gaussian mixture model (GMM) and power iteration clustering for clustering, FP-growth for frequent pattern mining, and block matrix abstraction for distributed linear algebra. Initial support has been added for model import/export in exchangeable format, which will be expanded in future versions to cover more model types in Java/Python/Scala. The implementations of k-means and ALS receive updates that lead to significant performance gain. PySpark now supports the ML pipeline API added in Spark 1.2, and gradient boosted trees and Gaussian mixture model. Finally, the ML pipeline API has been ported to support the new DataFrames abstraction.

Spark Streaming

Spark 1.3 introduces a new *direct* Kafka API (docs) which enables exactly-once delivery without the use of write ahead logs. It also adds a Python Kafka API along with infrastructure for additional Python API's in future releases. An online version of logistic regression and the ability to read binary records have also been added. For stateful operations, support has been added for loading of an initial state RDD. Finally, the streaming programming guide has been updated to include information about SQL and DataFrame operations within streaming applications, and important clarifications to the fault-tolerance semantics.

GraphX

GraphX adds a handful of utility functions in this release, including conversion into a canonical edge graph.

MLlib

Spark
MLlib

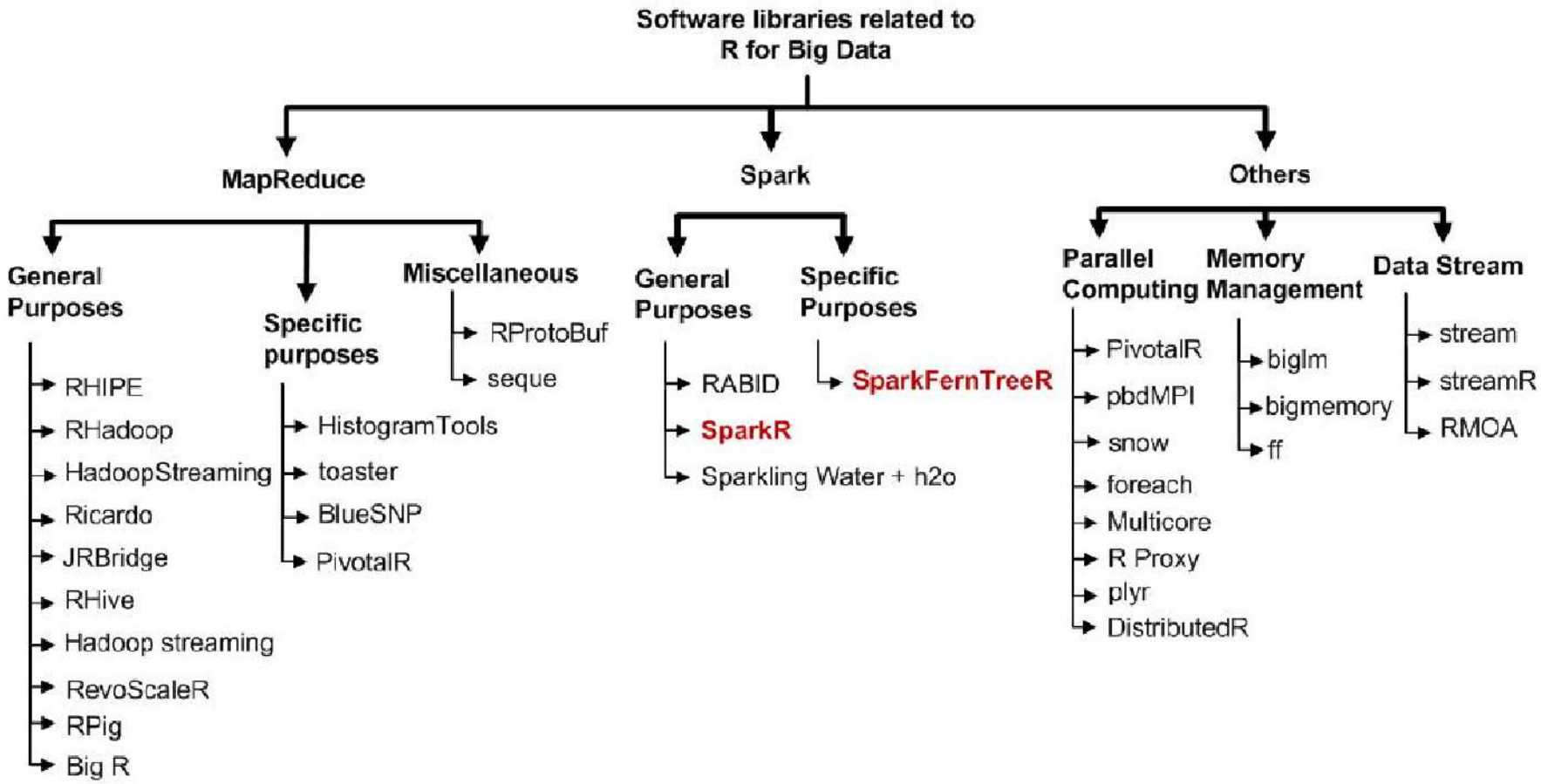
https:/ / spark.apache.org/ docs/ latest/ mllib-guide.html

Machine Learning Library (MLlib) Guide

MLlib is Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives, as outlined below:

- 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)

R for Big Data



Librería H_2O

H_2O

H_2O

http://0xdata.com/

Data Science in H_2O

- Cox Proportional Hazards Model
- Deep Learning
- Generalized Linear Model
- Gradient Boosted Regression and Classification
- K-Means
- Naive Bayes
- Principal Components Analysis
- Random Forest
- Summary
- Data Science and Machine Learning
- Stochastic Gradient Descent
- References

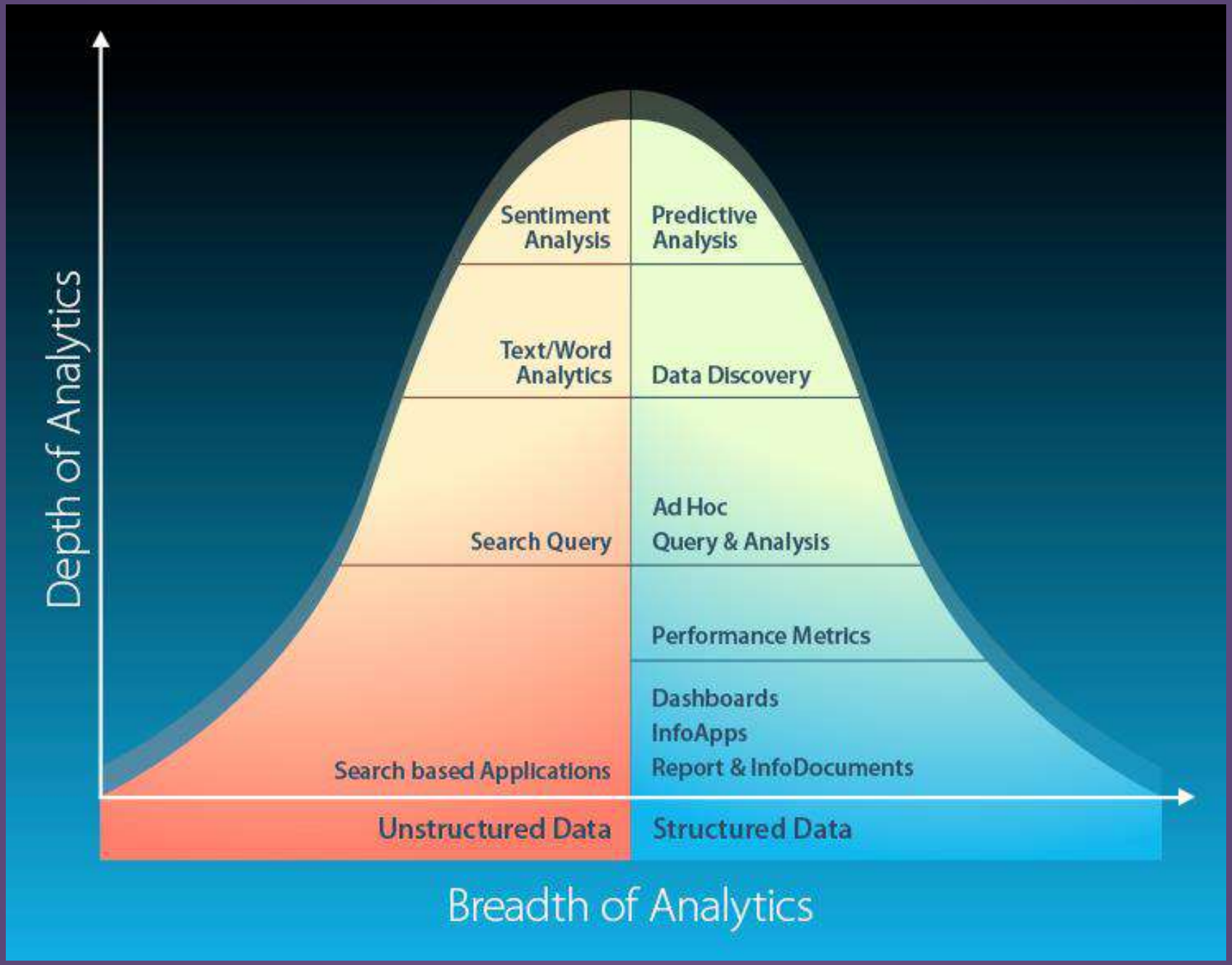
Soporte para R, Hadoop y Spark

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

- Librería que contiene algoritmos de Deep Learning
  - Récord del mundo en el problema MNIST sin preprocesamiento

http://0xdata.com/blog/2015/02/deep-learning-performance/

Algunas aplicaciones/ejemplos



http://www.informationbuilders.com/

Acceso y la búsqueda de información digital.

- **Los buscadores son eficientes, pero no eficaces.**
- **Ejemplos: sinonimia, veracidad (reputación), variedades diatópicas, operadores, tendencias…**

Acceso y la búsqueda de información digital.

- **Los buscadores son eficientes, pero no eficaces.**
- **Ejemplos: sinonimia, veracidad (reputación), variedades diatópicas, operadores, tendencias…**

Acceso y la búsqueda de información digital.

- **Los buscadores son eficientes, pero no eficaces.**
- **Ejemplos: sinonimia, veracidad (reputación), variedades diatópicas, operadores, tendencias…**

Acceso y la búsqueda de información digital.

- **Los buscadores son eficientes, pero no eficaces.**
- **Ejemplos: sinonimia, veracidad (reputación), variedades diatópicas, operadores, tendencias…**

Acceso y la búsqueda de información digital.

- **Los buscadores son eficientes, pero no eficaces.**
- **Ejemplos: sinonimia, veracidad (reputación), variedades diatópicas, operadores, tendencias…**

Acceso y la búsqueda de información digital.

- **Los buscadores son eficientes, pero no eficaces.**
- **Ejemplos: sinonimia, veracidad (reputación), variedades diatópicas, operadores, tendencias…**

"Búsqueda eficaz de información en la Web" (EDULP, 2011)

Internet y las redes sociales.

- **El nuevo reto de la Inteligencia Artificial en Internet y las redes sociales:**

  “asíncrono” vs. “síncrono”

  Reflexión/preparación vs. Inmediatez/visceralidad

- **¡Dimensión Humana!**
- **Análisis de Sentimientos.**

“Sentiment analysis: A review and comparative analysis of web services” (Information Sciences, 2015)

Reflexiones

- No hay cambios significativos de paradigmas en IA (T .S. Kuhn: La estructura de las revoluciones científicas)
  - Lógica Borrosa.
- Falta de comunicación Ciencias Cognitivas-Computación-HPC.

Reflexiones

- Escalabilidad en la complejidad de los algoritmos:
  - Límites de la computación.
  - NP completitud.
  - Ajedrez y juegos, Watson, Test de Turing…
  - IA y heurística.

UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

SMILe
Soft Management of Internet and Learning

Algunos retos para la Inteligencia Artificial en el siglo XXI

Dpto. de Tecnologías y Sistemas de Información
Universidad de Castilla-La Mancha
http://smile.esi.uclm.es

José A. Olivas
Joseangel.olivas@uclm.es

¡¡MUCHAS GRACIAS!!

La Plata, CACIC 2017