Grupo de Usuarios de R de Sevilla

Big Data con Spark y R

Manuel Chacón y Juan Andrés Tejero

4 Junio 2019, 19 h

Sala TIC4, CRAI Reina Mercedes

http://bit.ly/SevillaRmeetup

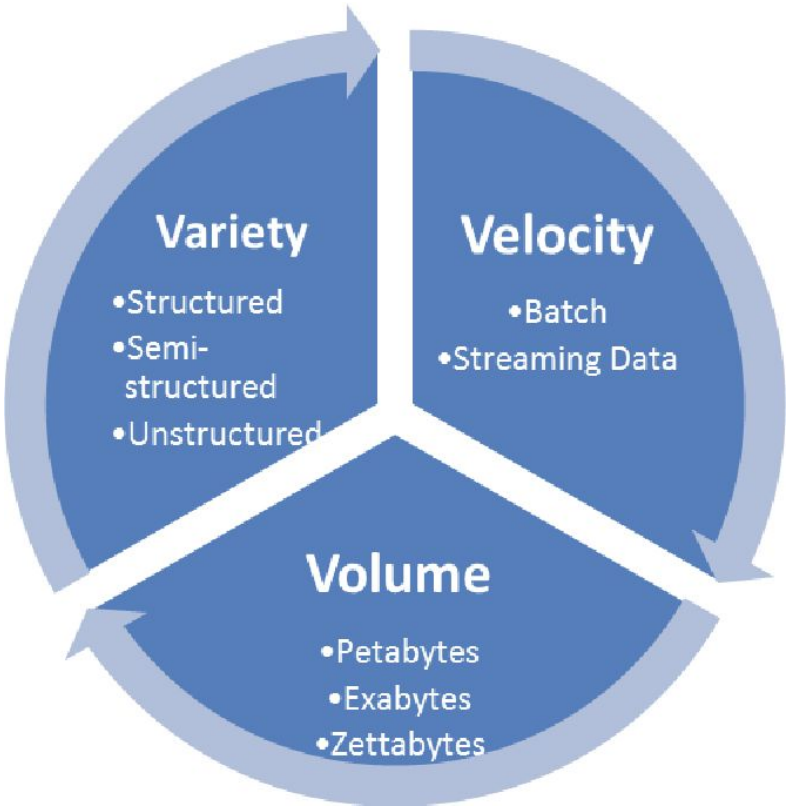
R consortium

DSC::Energy Analytics

Índice

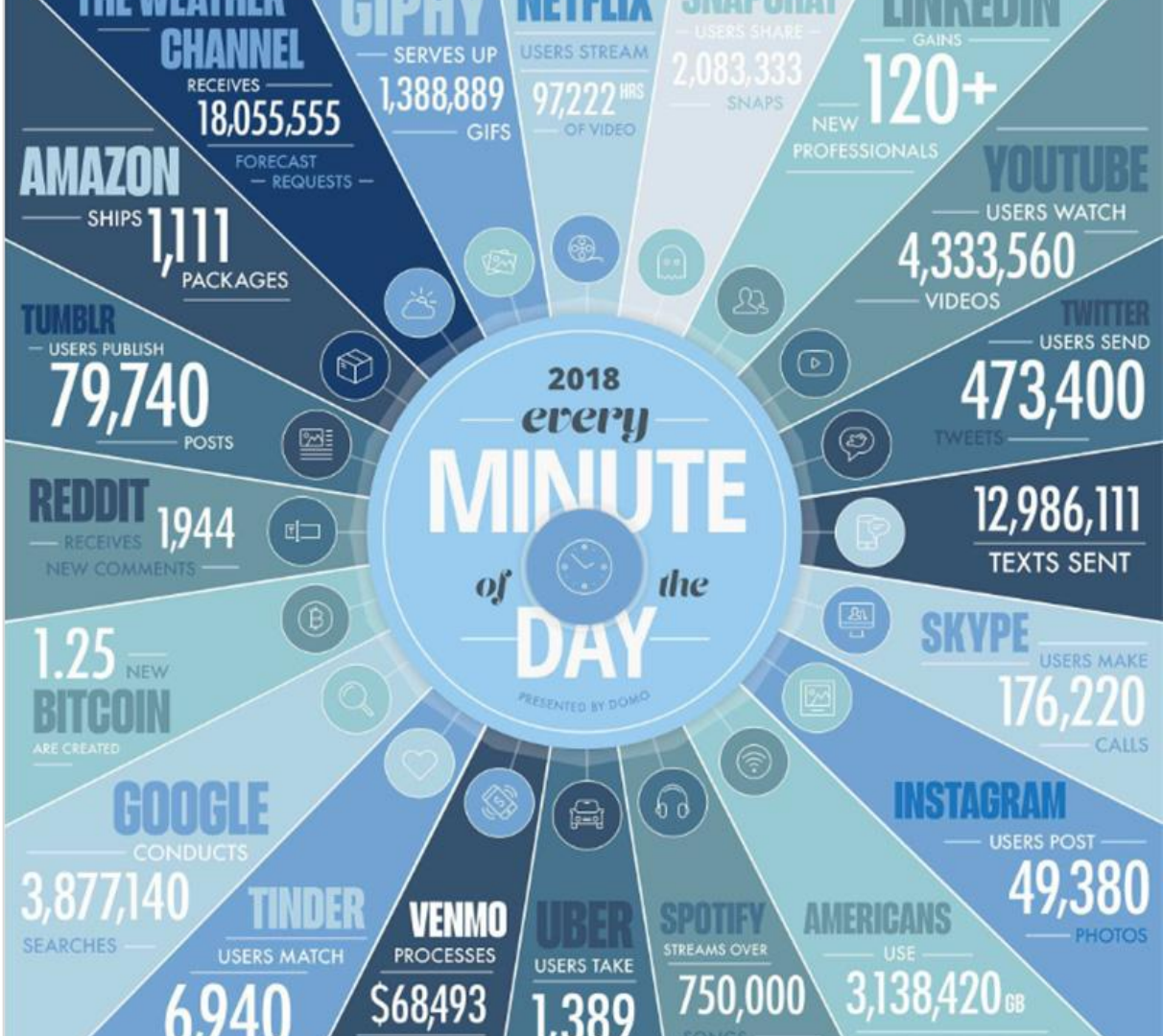
Crecimiento de generación de datos.

- Big data es aplicar herramientas de ciencia del dato para procesar grandes volúmenes de datos.
- Aumento de la generación y consumo de datos.



Volumen

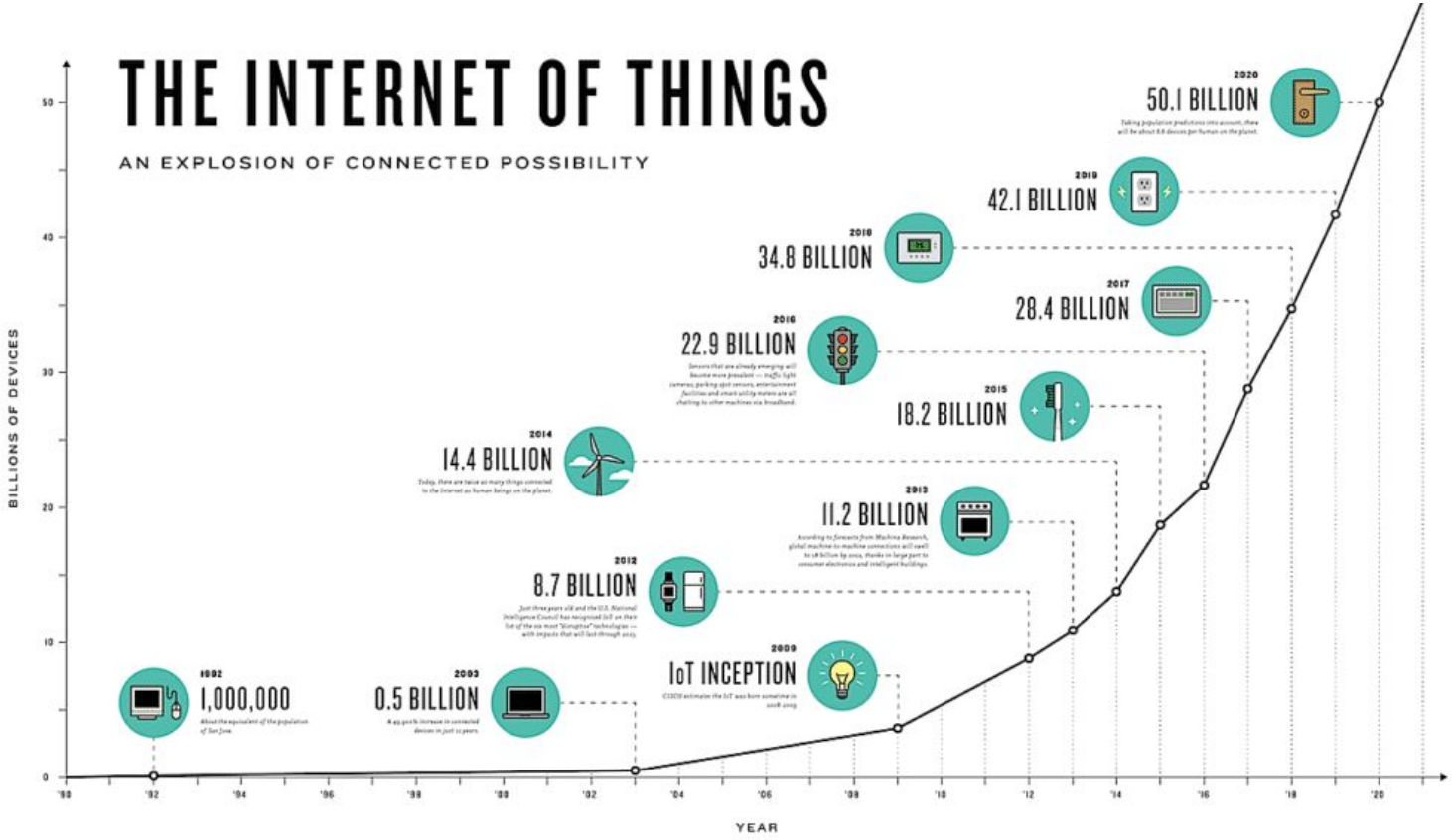
- Auge de la tecnología.
- Internet
- Dispositivos móviles.
- Más usuarios.



Velocidad

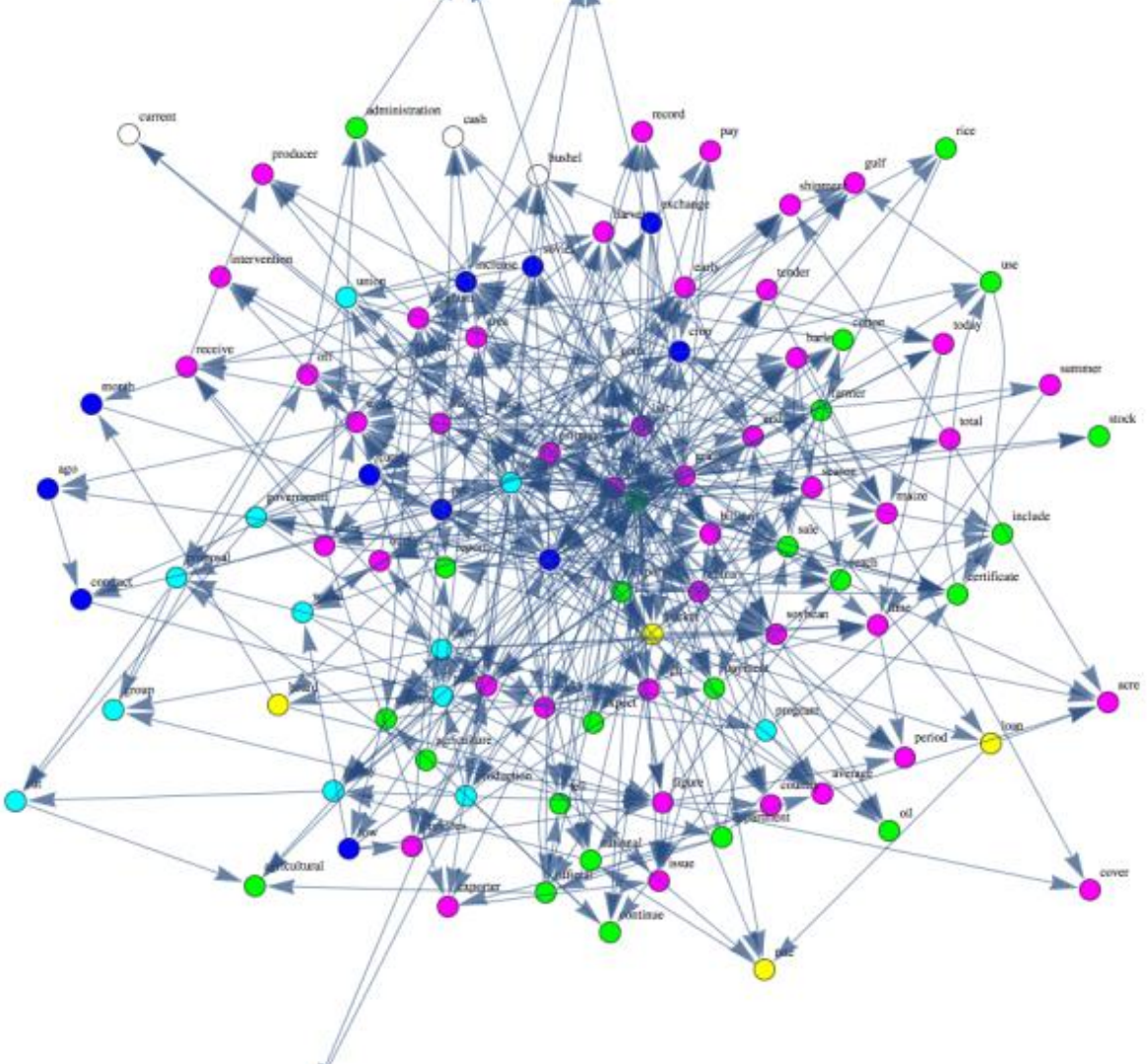
- Otra de la claves es la velocidad a la que los datos son generados y deben ser procesados.
- IoT, transacciones, imágenes, etc…
- Streaming

Velocidad


THE INTERNET OF THINGS
AN EXPLOSION OF CONNECTED POSSIBILITY
BILLIONS OF DEVICES
YEAR
1992
1,000,000
2003
0.5 BILLION
2009
IoT INCEPTION
2012
8.7 BILLION
2013
11.2 BILLION
2014
14.4 BILLION
2015
18.2 BILLION
2016
22.9 BILLION
2018
34.8 BILLION
2019
42.1 BILLION
2020
50.1 BILLION

Variedad

- Análisis de datos no estructurados:
  - Textos libres
  - Todo tipo de registros de actividad
  - Datos multimedia



¿Cuál es la revolución?

ANTES

- Sistemas centralizados
- Crecimiento vertical
- Una única máquina

VS

AHORA

- Sistemas distribuidos
- Crecimiento horizontal
- N máquinas
- Arquitectura Maestro/esclavo

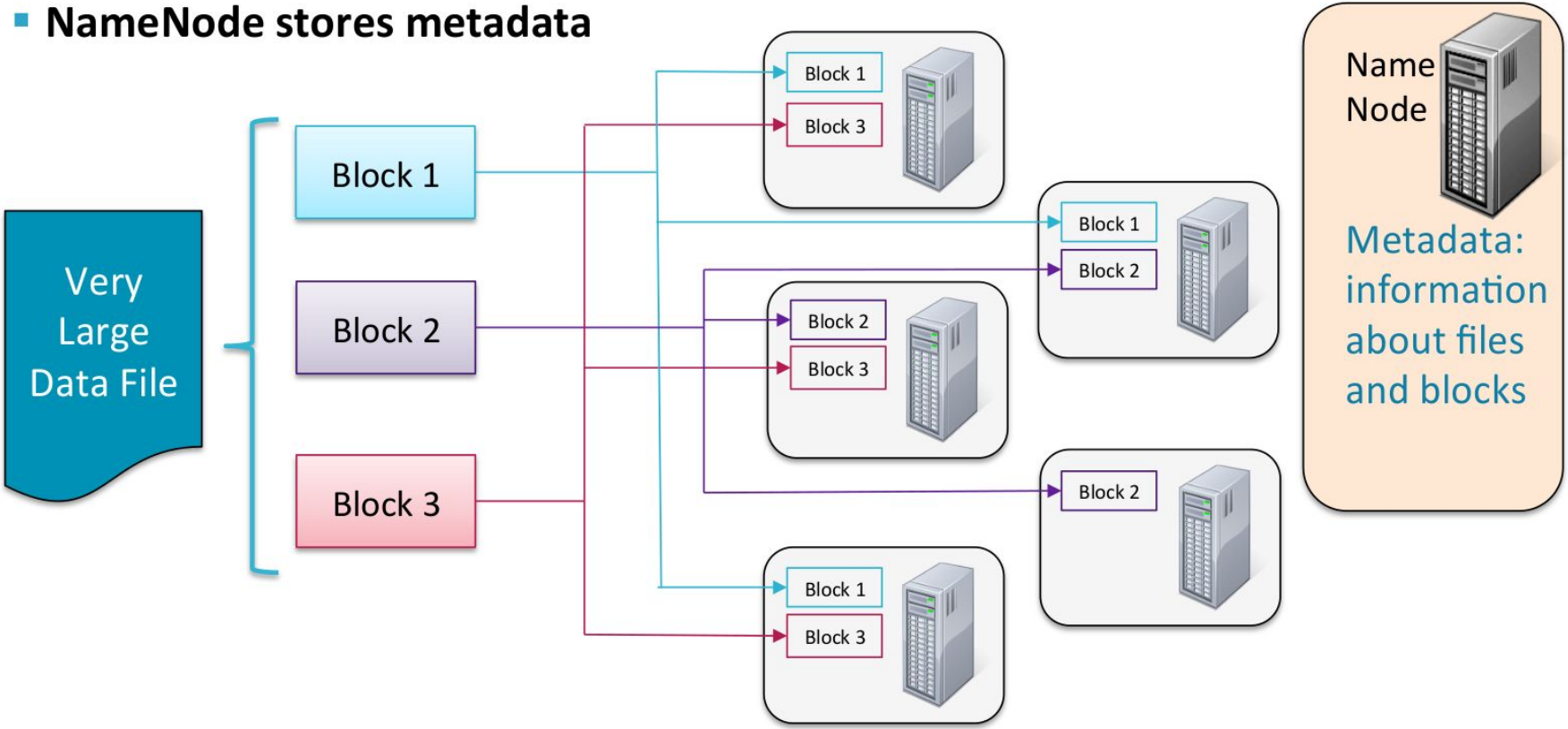
Hadoop

- Primera solución Big Data completa de propósito general.
- Lenguajes de consulta de alto nivel y transparentes.
- La comunicación entre nodos es mínima.
- Redundancia de datos, protección contra pérdida de datos.
- Tolerancia a fallos.

HDFS: Hadoop Distributed Filesystem

- HDFS almacena ficheros→ texto, binarios, comprimidos, etc…
- Cuando almacenamos un fichero en HDFS, este es dividido en bloques.
- Tiene un tamaño de bloque fijo configurable.
- Estos bloques serán las unidades de datos en tiempo de procesamiento.

- **Data files are split into blocks and distributed at load time**
- **Each block is replicated on multiple data nodes (default 3x)**
- **NameNode stores metadata**

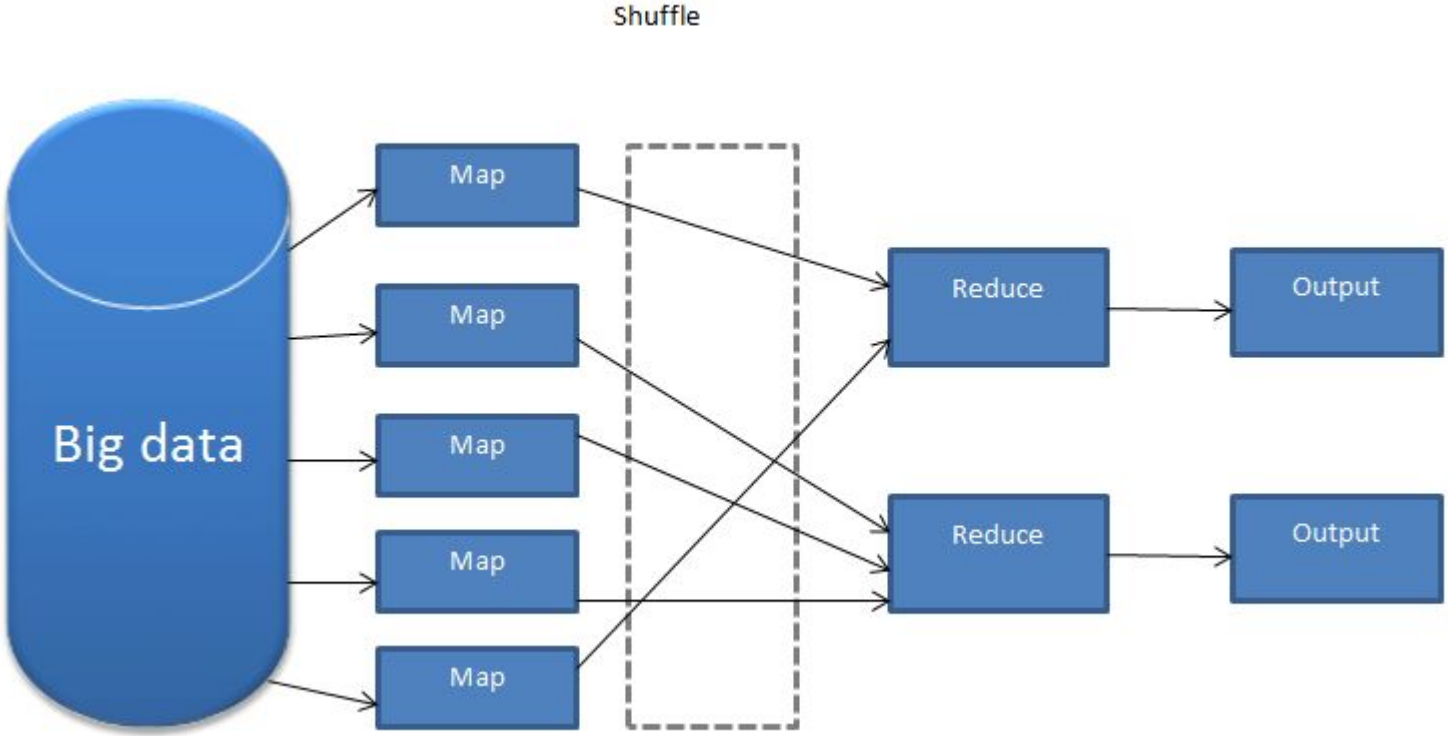


MapReduce

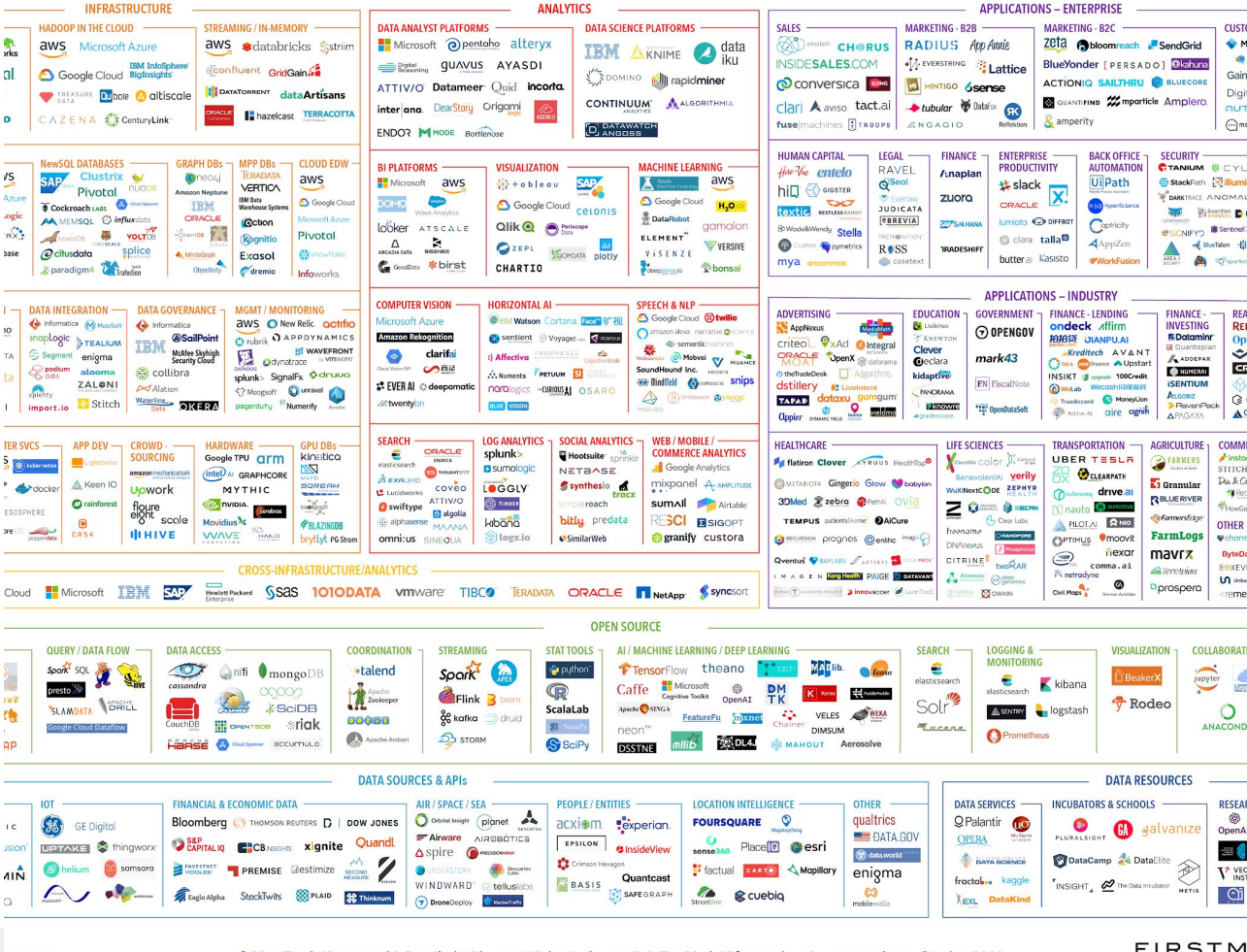
- Es un framework de programación que permite distribuir tareas entre nodos.
- El procesamiento siempre consta de dos fases Map y Reduce.
- Map → Dado un elemento, aplica una operación de transformación (mapeo).
- Reduce → Dado un conjunto de elementos, aplica una operación de agregación (reducción).
- Utiliza un modelos clave-valor.
- Es transparente para el programador, solo diseña las fases Map y Reduce. Igual para una que para 100 máquinas.

MapReduce

- Mapper
  - Cada tarea map actúa sobre un bloque de datos.
  - Se ejecuta donde se almacena el bloque.
- Shuffle
  - Organiza los datos originados como resultados de los mappers.
  - Esta fase tiene lugar una vez que todos los mappers han terminado y antes de que se inicien los reducers.
- Reducer
  - Opera sobre la salida (ordenada) de los mappers.
  - Aquí se general el resultado final de la tarea.


Shuffle
Big data
Map
Map
Map
Map
Map
Reduce
Reduce
Output
Output

BIG DATA & AI LANDSCAPE 2018

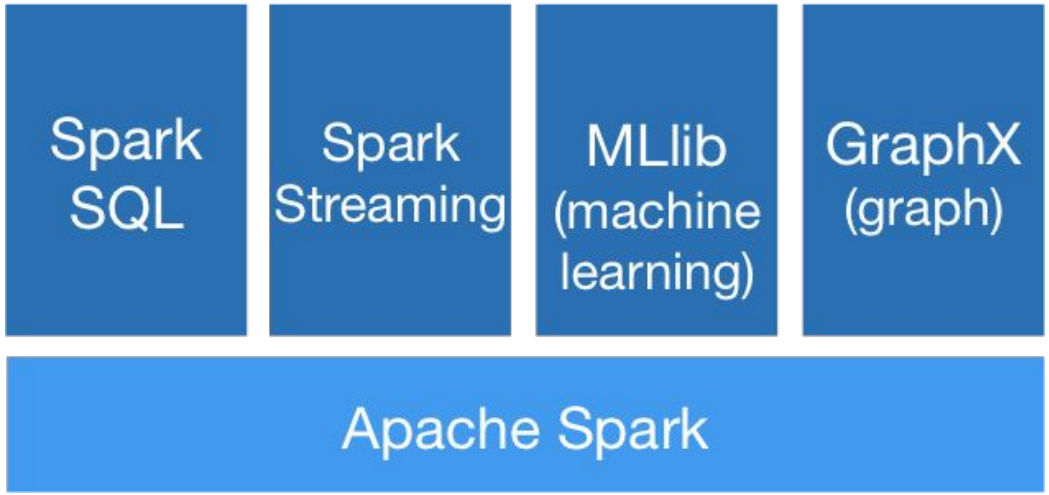


http://mattturck.com/wp-content/uploads/2018/07/Matt_Turck_FirstMark_Big_Data_Landscape_2018_Final.png

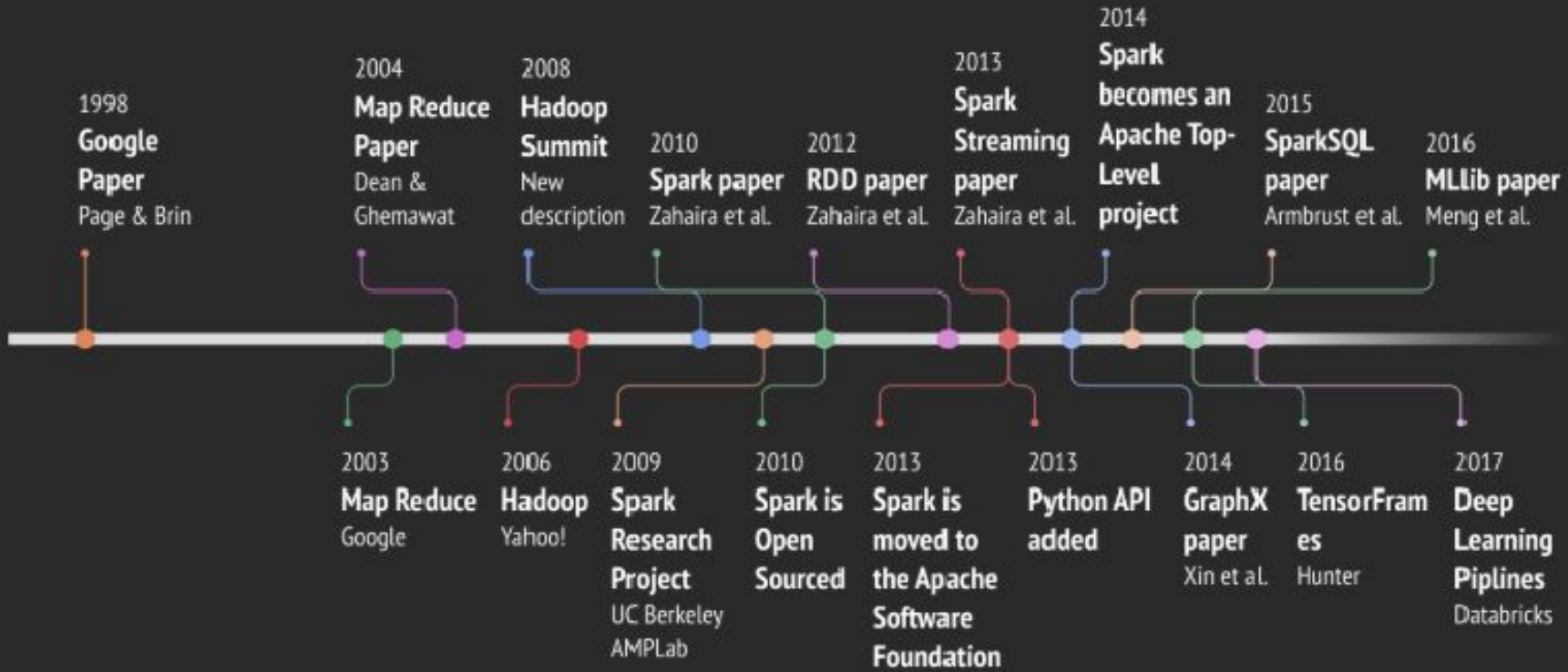
Apache Spark

Framework open-source para el procesamiento distribuído en memoria de datos

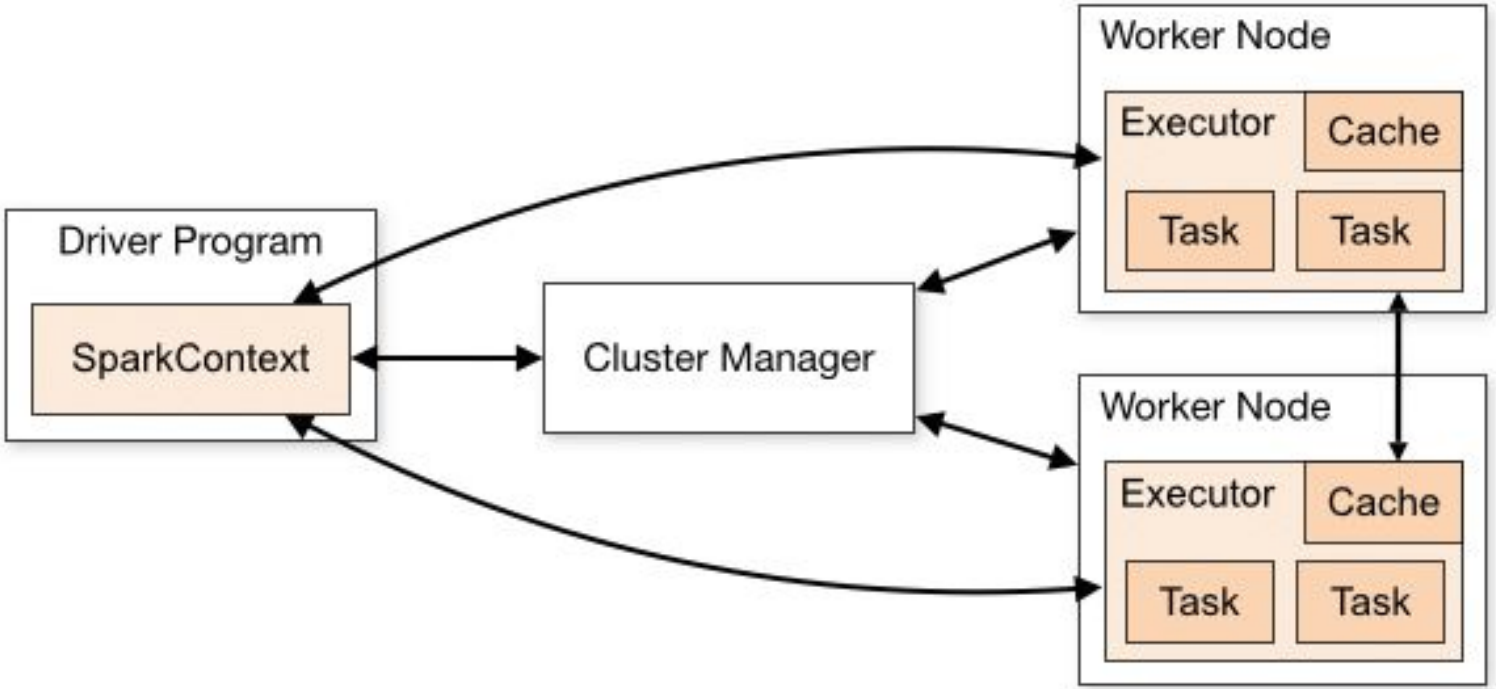
- Escrito en Scala → APIs para Scala, Java, Python … y R
- Desarrollado por el AMPLab (Berkeley, CA) → Comunidad Open-Source



Apache Spark Timeline



Apache Spark: Funcionamiento



Apache Spark: Funcionamiento - Estructura datos

History of Spark APIs

Distribute collection of JVM objects

Functional Operators (map, filter, etc.)

Distribute collection of Row objects

Expression-based operations and UDFs

Logical plans and optimizer

Fast/efficient internal representations

Internally rows, externally JVM objects

Almost the "Best of both worlds": type safe + fast

But slower than DF
Not as good for interactive analysis, especially Python

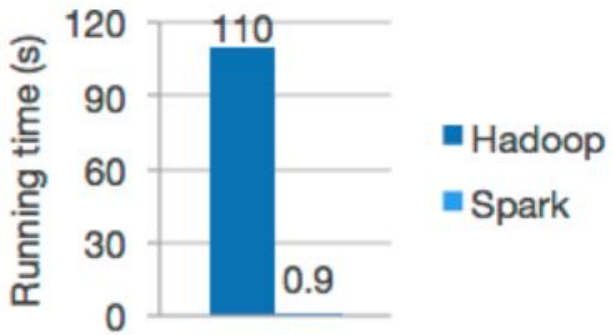
databricks

Apache Spark: Funcionamiento - Memoria/Disco

Puede ejecutarse tanto en memoria como en disco

- x10 de velocidad en disco
- x100 de velocidad en memoria

vs. Hadoop

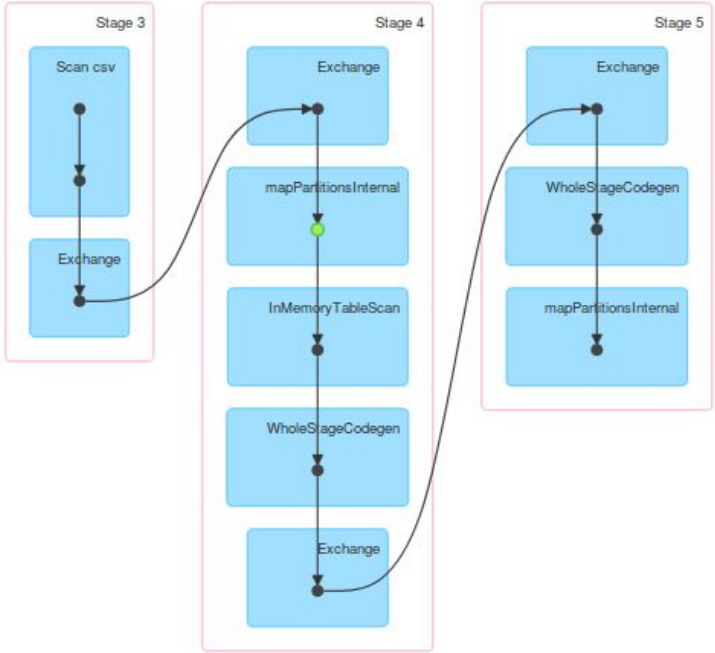


Logistic regression in Hadoop and Spark

Apache Spark: Funcionamiento - DAGs

- Operaciones en Spark
- Lazy evaluation

transformación
acción



Apache Spark: Funcionamiento - Modos

Standalone

→ Incluído con Spark. Simple y fácil de configurar

Apache Mesos

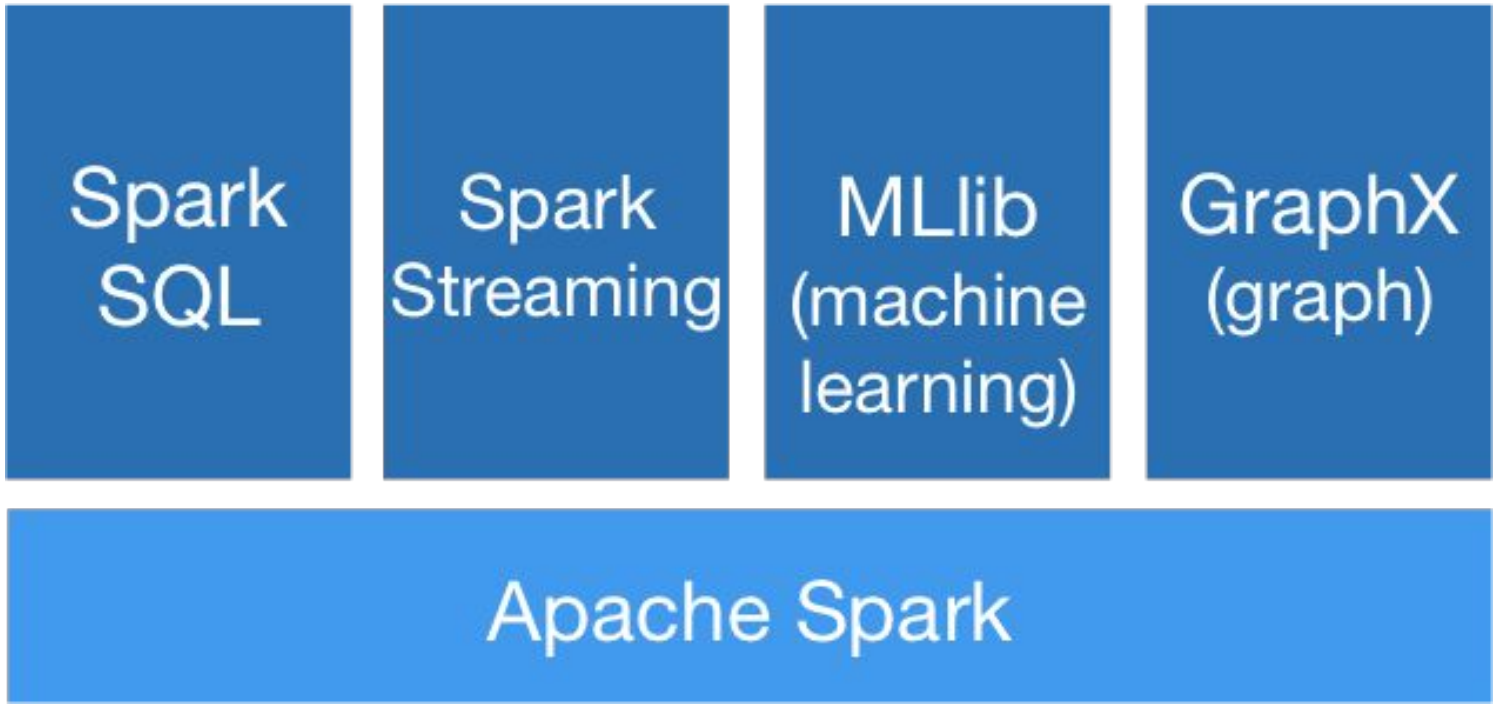
→ RM de propósito general para datacenters

Hadoop YARN

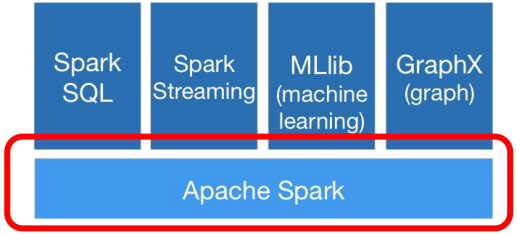
→ RM incluído con Hadoop v2. Óptimo si se usa HDFS

Kubernetes

Apache Spark: Módulos



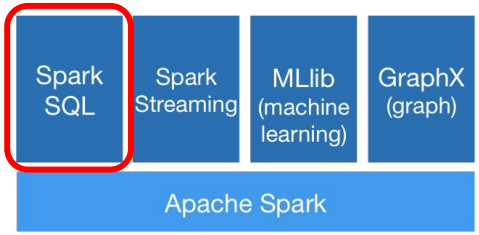
Apache Spark: Módulos



Spark Core

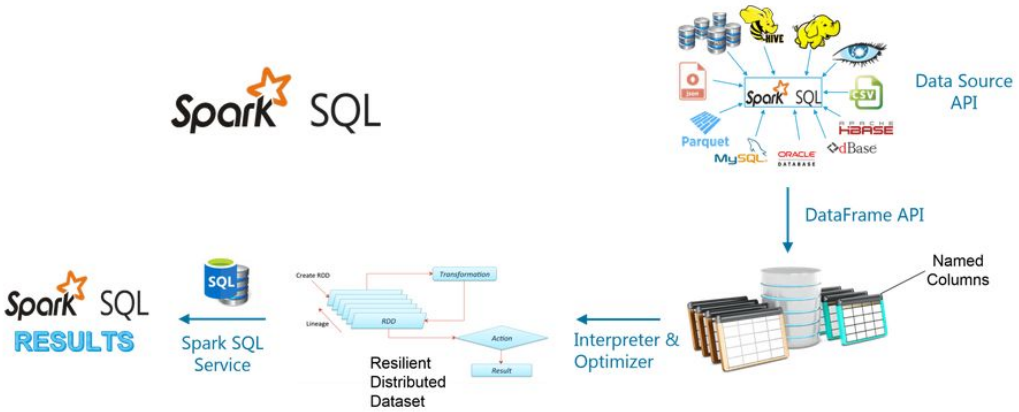
- Funcionalidades básicas de spark
  - Scheduling
  - Gestión de memoria
  - Gestión de errores
  - Operaciones E/S
  - RDDs y operaciones básicas
    - Map, filter, reduce

Apache Spark: Módulos

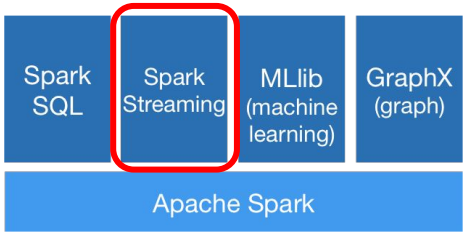


Spark SQL

- Herramienta para trabajar con datos estructurados con SQL (HQL)
- Múltiples fuentes de datos
  - Hive, Parquet, ORC, JSON…
- Permite combinar consultas básicas SQL con analítica compleja

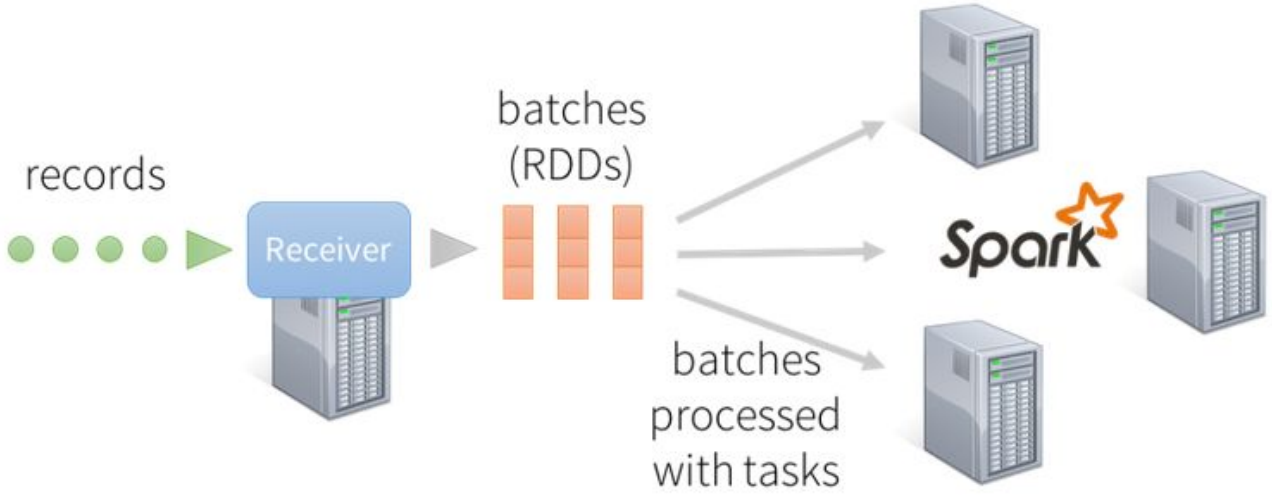


Apache Spark: Módulos

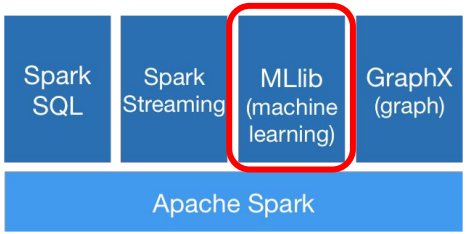


Spark Streaming

- Módulo para el procesamiento en tiempo real de datos
- Micro-batching → Apache Flink



Apache Spark: Módulos



MLlib: Librería de ML nativa de Spark

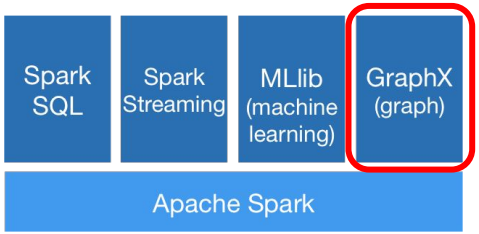
- Ejecución de manera distribuida
- LR, Naive Bayes, GLM, Random Forest, GBT, ALS, K-means…
- Catálogo limitado → Necesidad de usar librerías de terceros

H20 → Sparkling Water

MMLlib → LightGBM, CNTK

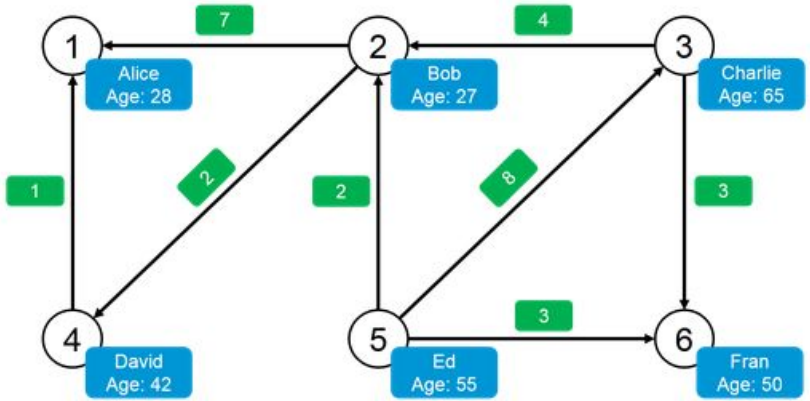
Tensorflow on Spark

Apache Spark: Módulos



GraphX

- Extensión para trabajar con datos estructurados en forma de grafos
  - RRSS, redes, transporte, sistemas de recomendaciones, problemas geoespaciales...
  - RDD → Resilient Distributed Property Graph (RDPG)
- Grafos múltiples dirigidos
  - Permiten la ejecución en paralelo de acciones
- Operaciones especiales sobre grafos
  - joinVertices, subgraph, aggregateMessages...



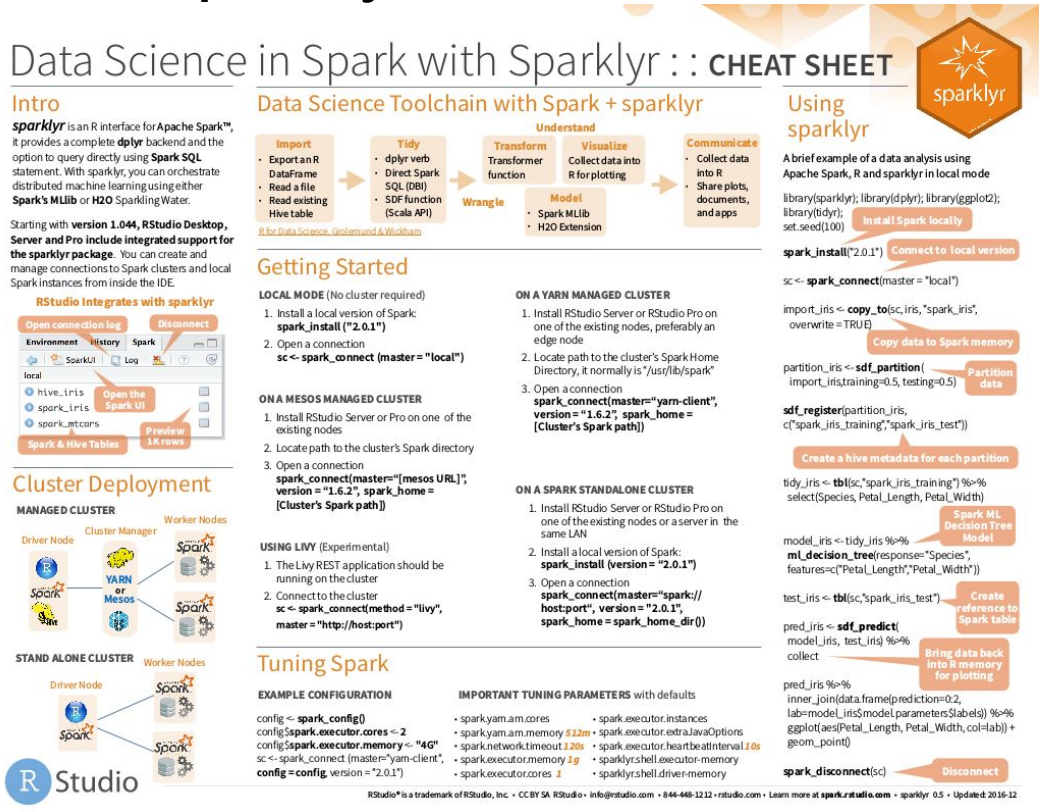
Apache Spark: SparkR

- Paquete oficial de Spark para R
- Desarrollado inicialmente por el AMPLab → Open-source
  - Integración con el proyecto Apache Spark desde la v1.4
- Bastantes limitaciones
  - Spark Streaming y GraphX aún no implementados
  - No soporta conexión con aplicaciones de terceros
  - No soporta modo yarn-cluster
  - Data.table… :(


WELCOME TO GGPLOT2 Unit
ggplot2
tidyverse
FEAT. MARKOV CHAIN MC
Wicked Wickham

Apache Spark: Sparklyr

Apache Spark: Sparklyr



https://github.com/rstudio/cheatsheets/raw/master/sparklyr.pdf

Apache Spark: SparkR vs. Sparklyr

- Integrado con los principales paquetes de R: dplyr, DBI, broom...
- Soporta ML Pipelines y procesamiento de grafos (GraphX)
- Compatible con Livy y YARN
- Es posible exportar modelos de ML a Java usando MLeap
- Integración con RStudio y RStudio Server
- Compatible con extensiones Scala
- Soporta extensiones de terceros: H2O, SAS data...

Apache Spark: SparkR vs. Sparklyr

Feature	SparkR	sparklyr
Data input & output	+ +	+ +
Data manipulation	-	+ + +
Documentation	+ +	+ +
Ease of setup	+ +	+ +
Function naming	- -	+ + +
Installation	+	+ +
Machine learning	+	+ +
Range of functions	+ + +	+ +
Running arbitrary code	+	+ +
Tidyverse compatability	- - -	+ + +

https://eddjberry.netlify.com/post/2017-12-05-sparkr-vs-sparklyr/


IT'S
DEMO TIME!
memegenerator.es

Conclusiones

- Spark nos permite procesamiento en memoria distribuído de Big Data
  a. Streaming, graph processing, SQL y ML
  b. Tener en cuenta sobrecarga de dividir datos
- Usable desde R
  a. SparkR → aún verde
  b. Sparklyr → más maduro y con compatibilidad con Tidyverse, H2O...
- Aún limitaciones en R
  a. Necesidad de interconexión con librerías de ML (TF on Spark, MMLlib, scikit-learn...)

Grupo de Usuarios de R de Sevilla

Big Data con Spark y R

Manuel Chacón y Juan Andrés Tejero

4 Junio 2019, 19 h

Sala TIC4, CRAI Reina Mercedes

http://bit.ly/SevillaRmeetup

R consortium

DSC::Energy Analytics