

Sistemas Paralelos y Distribuidos

Máster en Ciencia y Tecnología Informática

Diseño de Sistemas Distribuidos

Máster en Ingeniería Informática

Curso 2021-2022

Sistemas escalables en entornos distribuidos

Alejandro Calderón Mateos,

acaldero@inf.uc3m.es

Óscar Pérez Alonso,

oscar@lab.inf.uc3m.es

Félix García Carballeira

fgcarball@inf.uc3m.es

Contenidos

Big
Big
Big
Big
Big

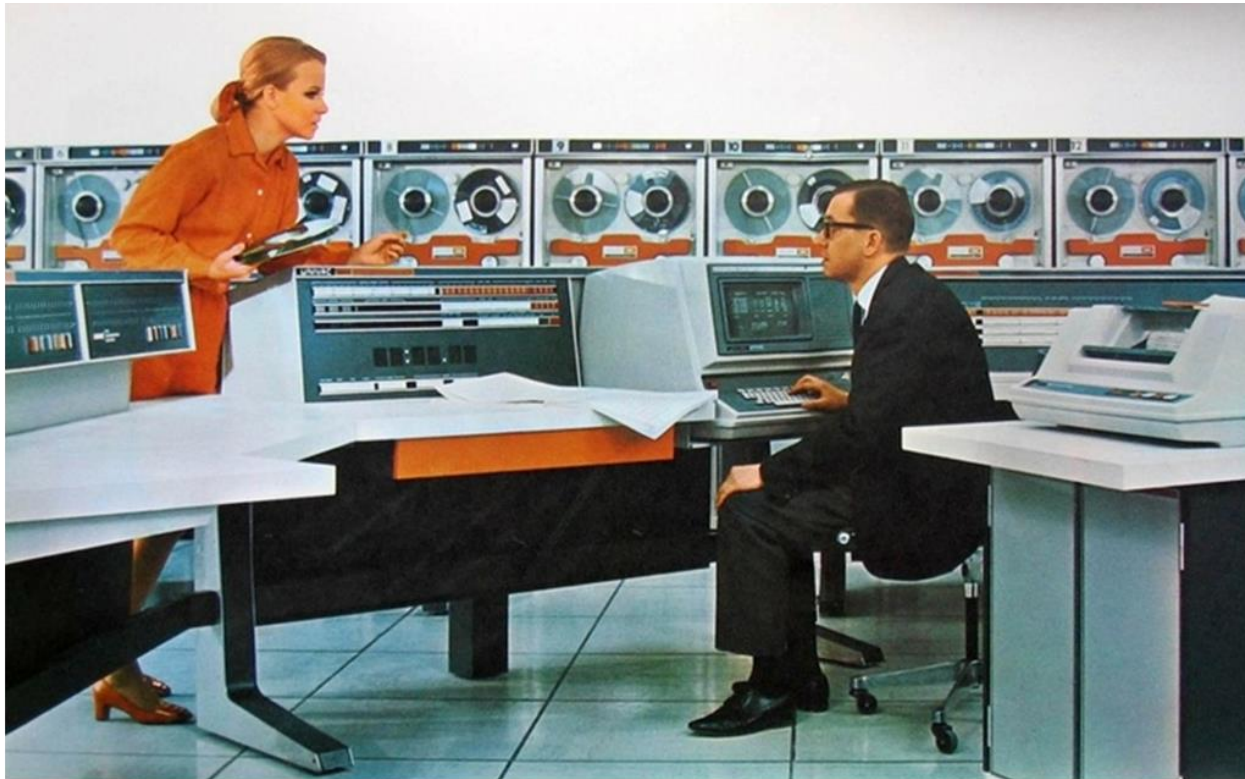
— Inicios

Los inicios: sistemas **no** distribuidos...



Centralizado

Los inicios: sistemas centralizados...



~ 1960

Los inicios: sistemas centralizados...



- Sistema centralizado
 - Mejor mantenimiento
 - Compartición de recursos
 - Repartición de costes

~ 1960

Los inicios: ARPANET...

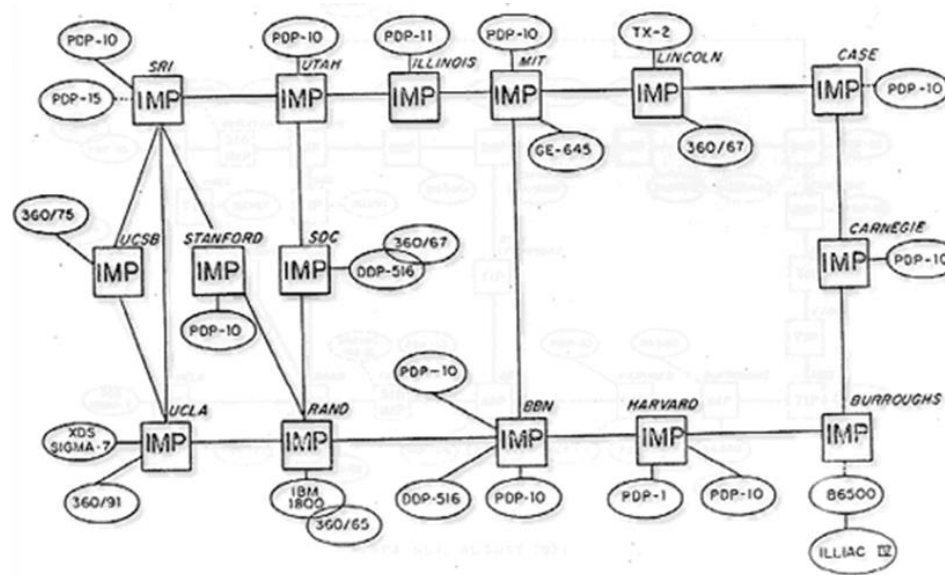


ARPANET

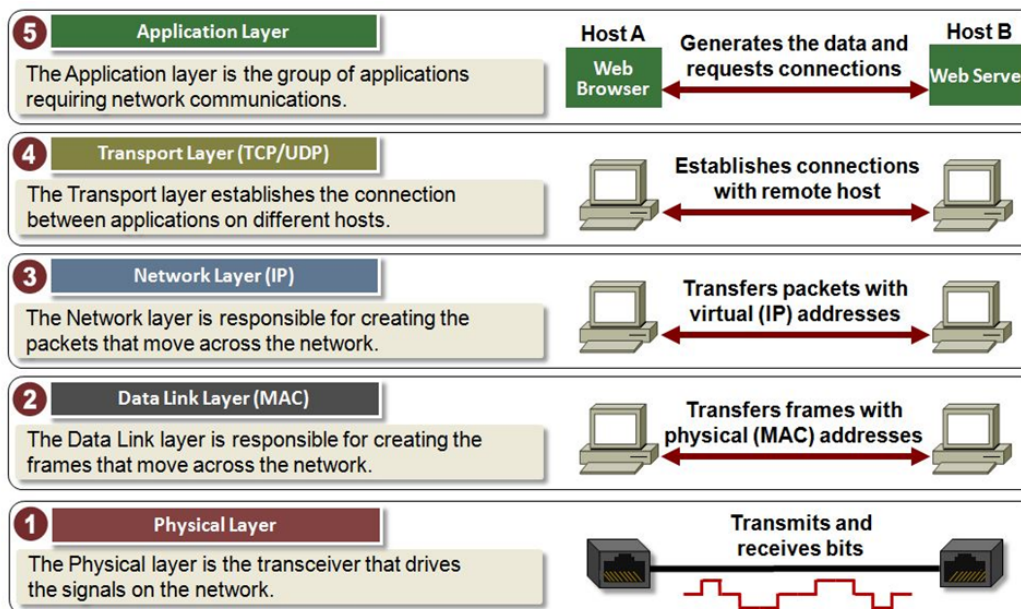


~ 1969

Los inicios: pre-Internet...

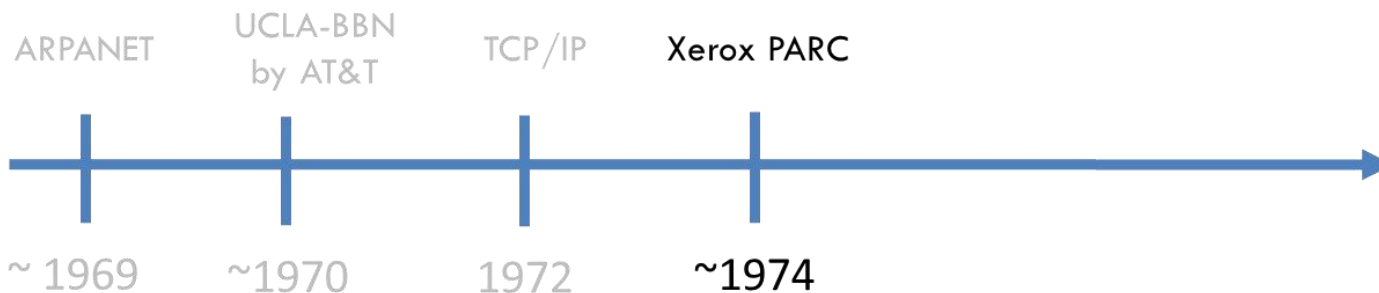
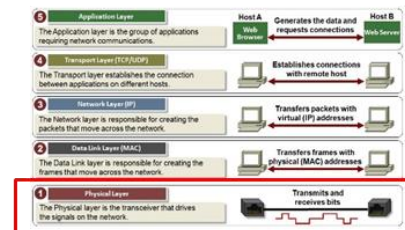


Los inicios: TCP/IP...

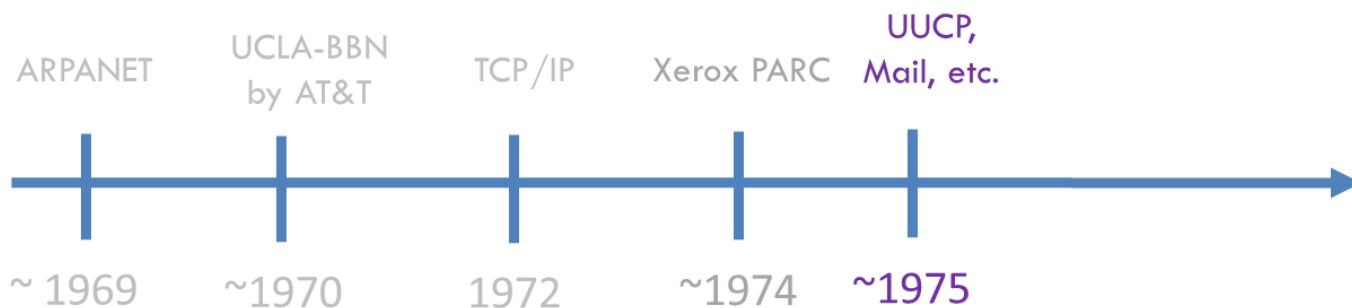
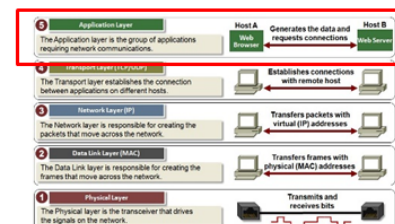
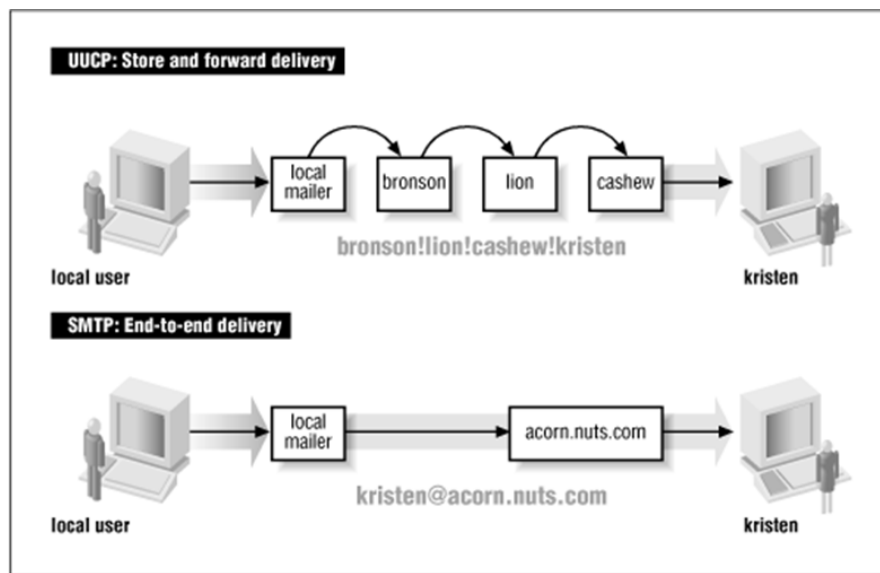


Los inicios: Ethernet...

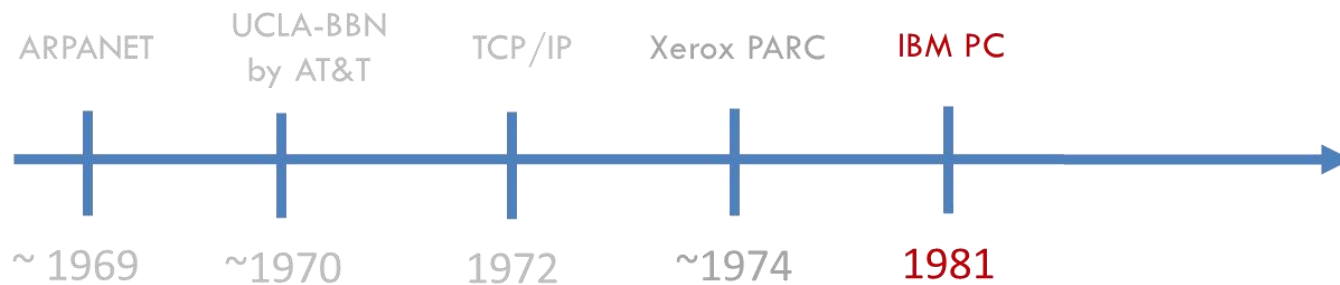
Pin	Signal ID	T568A Color	T568B Color	Pins on plug face (socket is reversed)
1	TX+	 white/green stripe	 white/orange stripe	
2	TX-	 green solid	 orange solid	
3	RX+	 white/orange stripe	 white/green stripe	
4	7 - 30VDC	 blue solid	 blue solid	
5	7 - 30VDC	 white/blue stripe	 white/blue stripe	
6	RX-	 orange solid	 green solid	
7	GROUND	 white/brown stripe	 white/brown stripe	
8	GROUND	 brown solid	 brown solid	



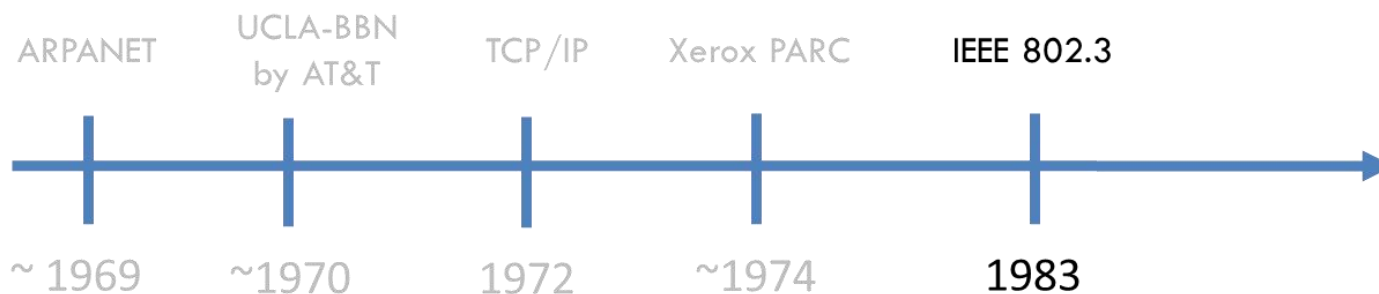
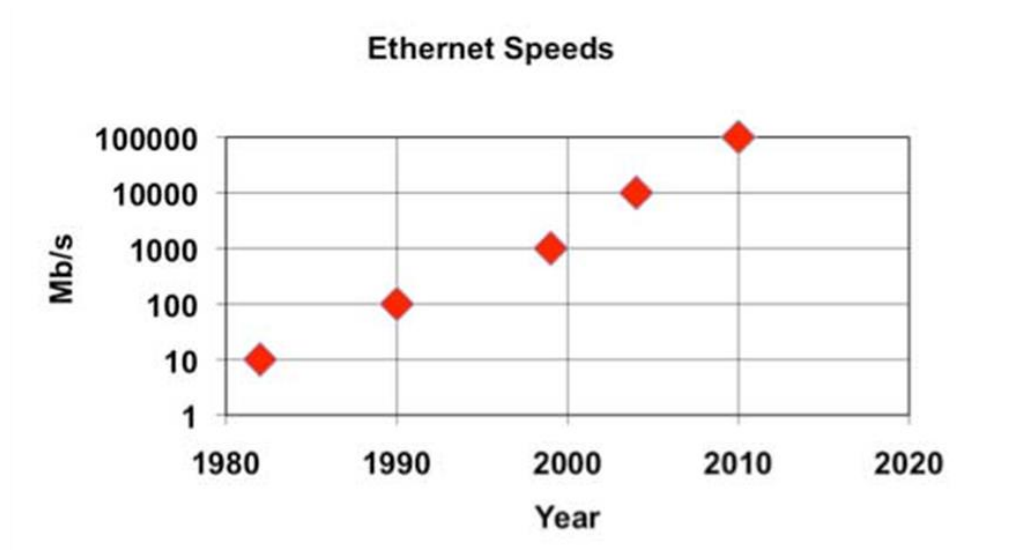
Los inicios: primeros servicios en red...



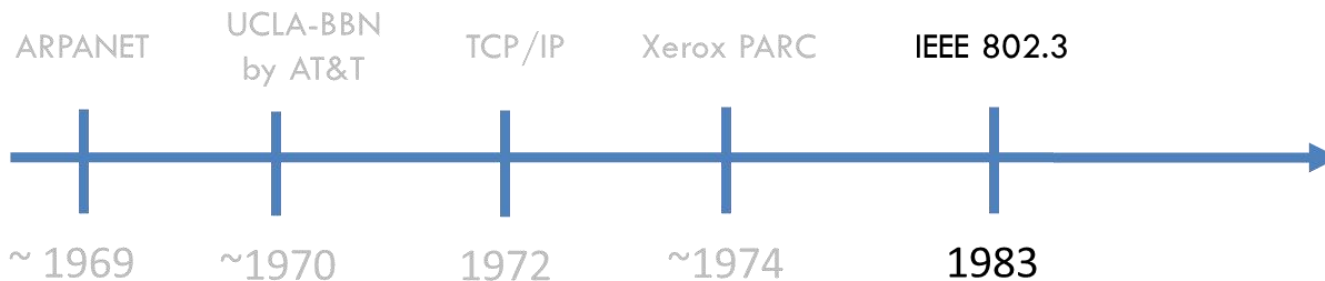
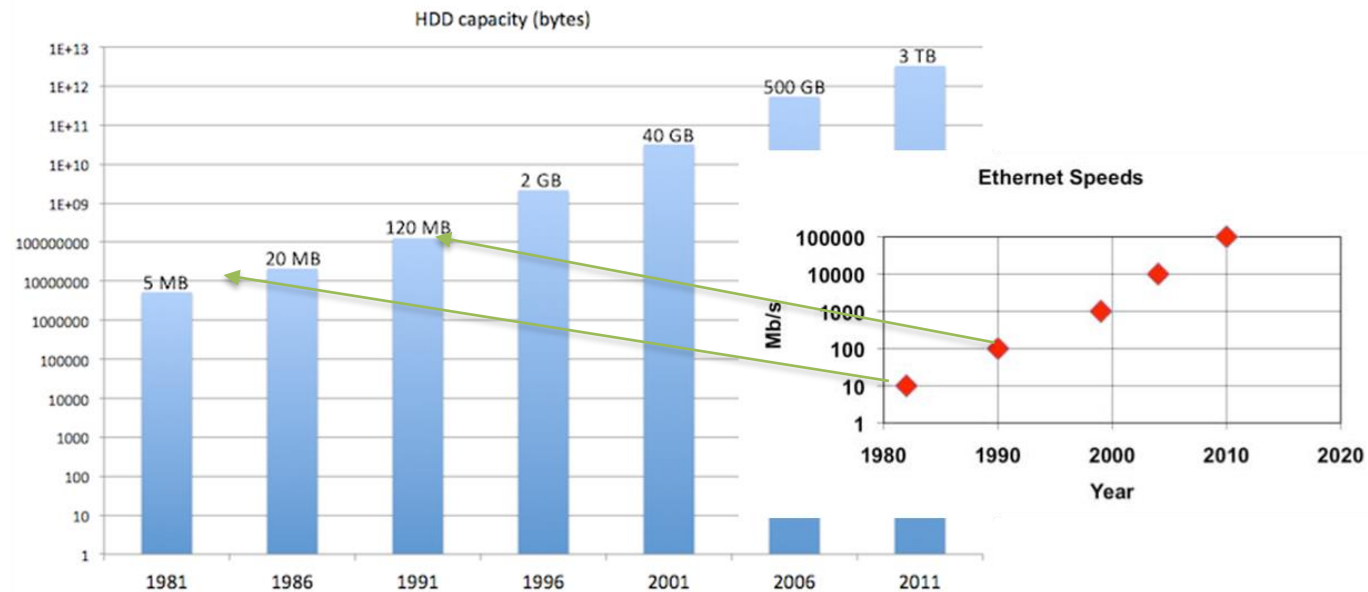
Los inicios: un ordenador en cada...



Los inicios: Ethernet más rápida...



Los inicios: un disco duro en segundos...



Los inicios: compartir información...

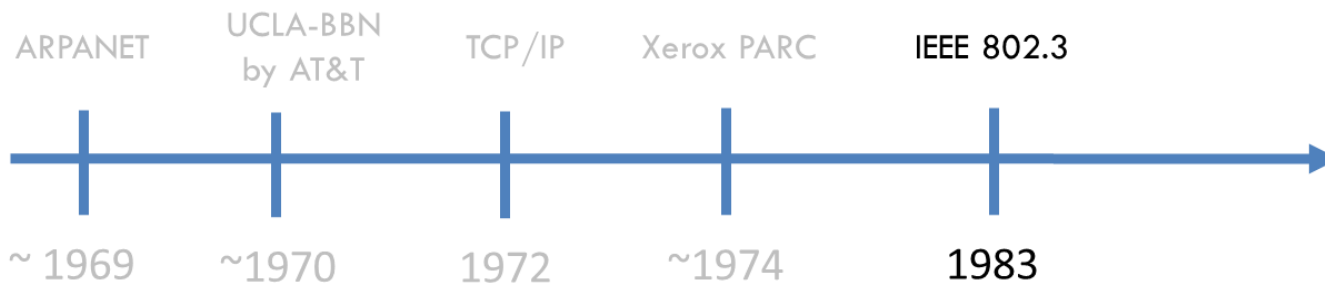


```
File Area [1-31], [L]ist, [H]elp, [Q]uit, [ENTER = ALL]: [1]

1.  New Uploads
2.  Special Downloads
3.  Communications
4.  Internet Utilities
5.  TriBBS Utilities
6.  BBS Software
7.  BBS Utilities
8.  Anti-Virus
9.  File Compression
10. Text Files
11. Word Processors/Editors
12. Windows 95/NT Utilities
13. DOS Utilities
14. Windows 3.1 Utilities
15. Sound/Audio/Music Utilities
16. Graphics/Image Utilities
17. Games [Windows]
18. Games [DOS]
19. Game Levels [HABDs etc..]
20. Online BBS Doors/Games
21. LORD/LORD II Utilities
22. Programming Tools
23. Business Utilities
24. Home/Finance
25. Self-Improvement
26. Educational/Childrens Programs
27. Miscellaneous Other
28. Wildcat 5 Utilities/Games
29. USA PalmPilot Utilities
30. Christian Interest
31. File Uploading Area

- End of File Area List -
[1-31], [L]ist, [A]ll, [S]ort, [F]ind, [H]elp, [Q]uit, [ENTER = NEXT]: [ ]
```

Computer network (& net. apps.):
computers are explicitly visible



Contenidos

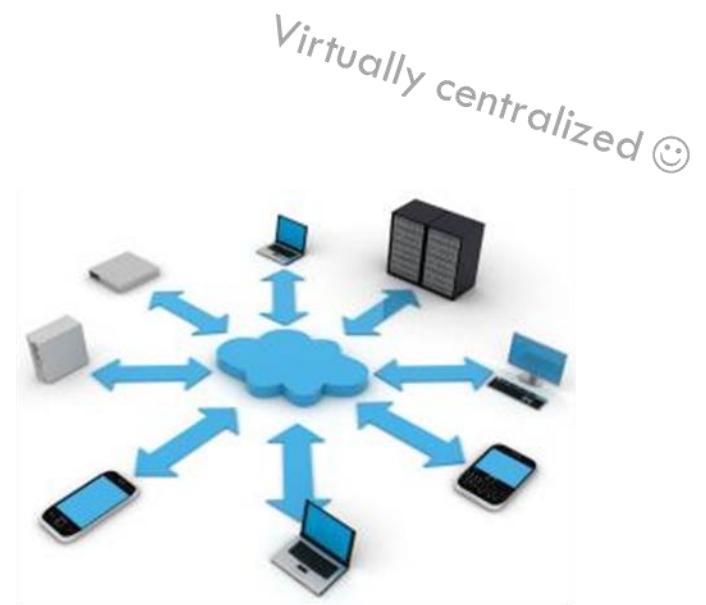
Big
Big
Big
Big
Big

- Inicios
- Transición

Transición...



Centralized



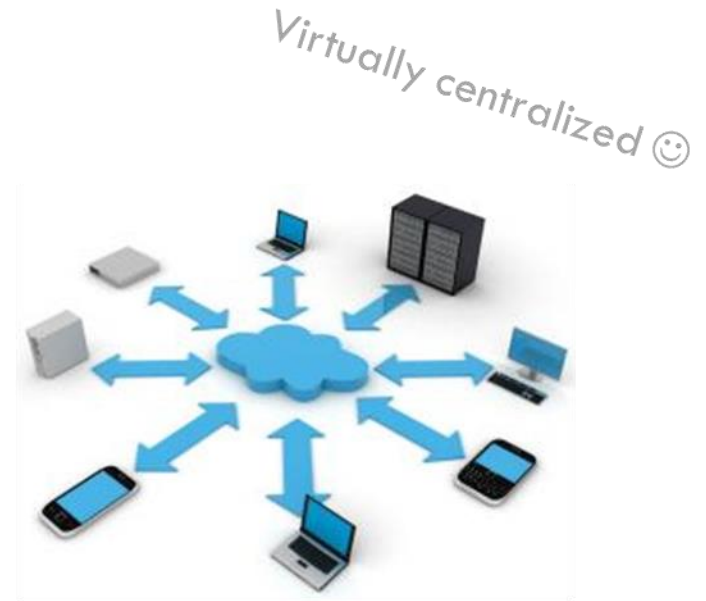
Distributed

Distributed system:
existence of multiple elements is transparent

Las transiciones no son fáciles...



Centralized

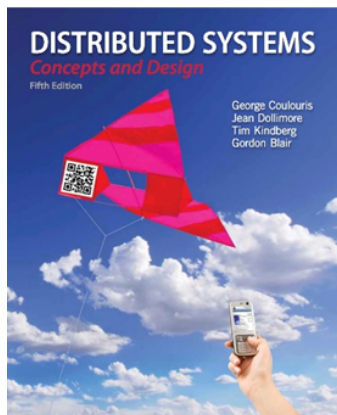


Distributed

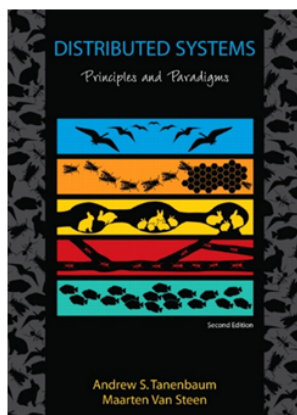
Distributed system:
existence of multiple elements is transparent

Las transiciones no son fáciles...

Expectativas



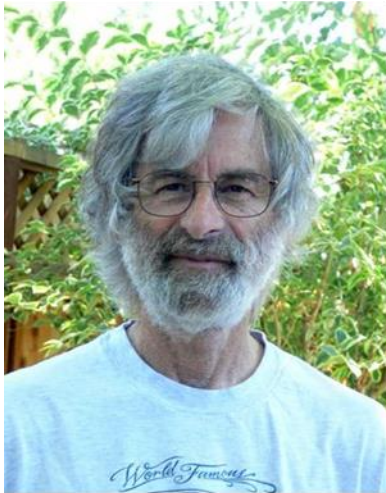
“We define a distributed system as one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages”



“A distributed system is a collection of independent computers that appears to its users as a single coherent system”

Las transiciones no son fáciles...

Realidades



- “A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable”

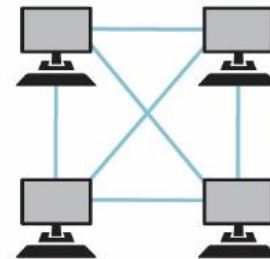
Leslie Lamport

Las transiciones no son fáciles...

Retos...

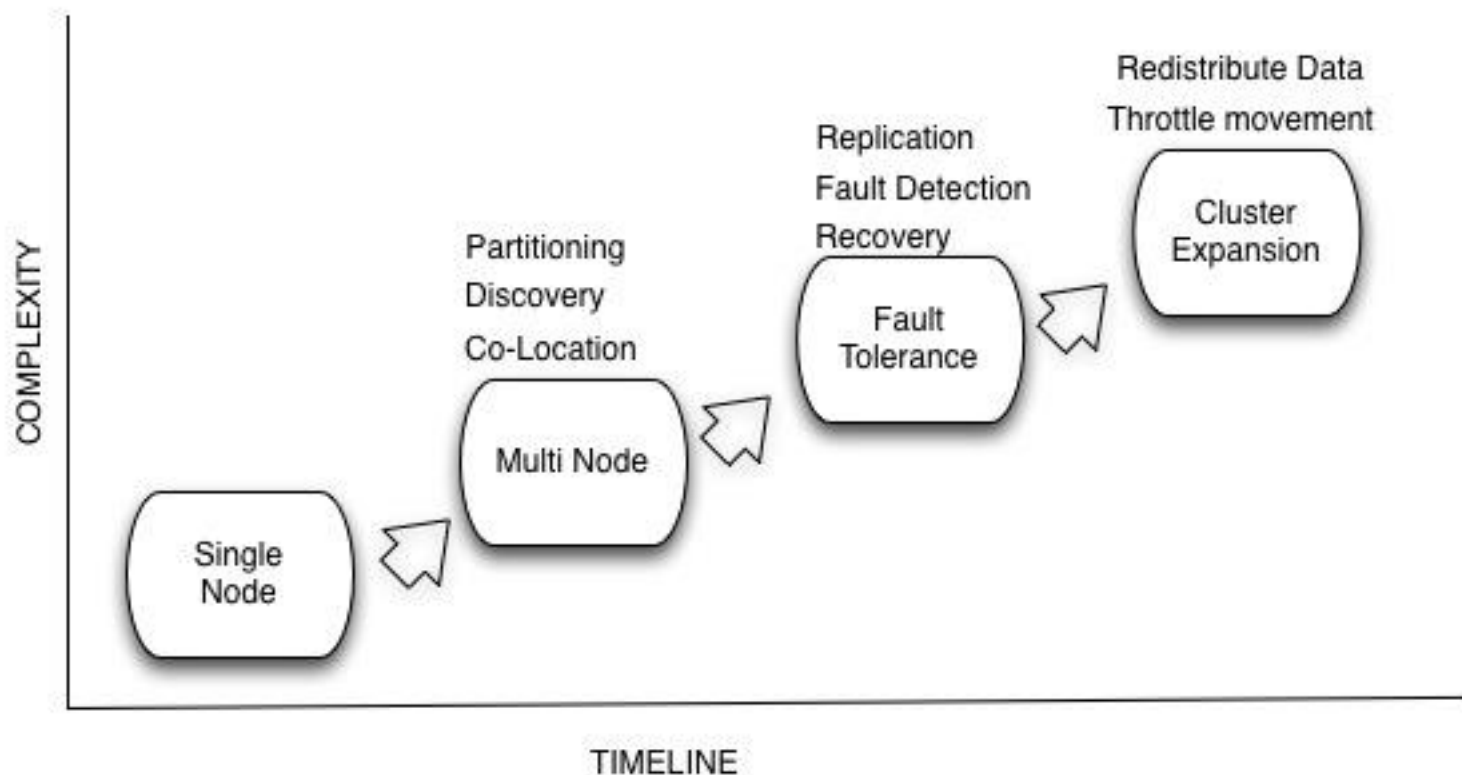
**Peter
Deutsch
et al.**

- ☐ The network is reliable
- ☐ The network is secure
- ☐ The network is homogeneous
- ☐ The topology does not change
- ☐ Latency is zero
- ☐ Bandwidth is infinite
- ☐ Transport cost is zero
- ☐ There is one administrator



Las transiciones no son fáciles...

Retos...

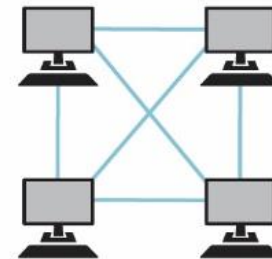


Las transiciones no son fáciles...

Áreas de trabajo...



- ☐ Heterogeneity
- ☐ Openness
- ☐ Security
- ☐ Scalability
- ☐ Failure Handling
- ☐ Concurrency
- ☐ Transparency

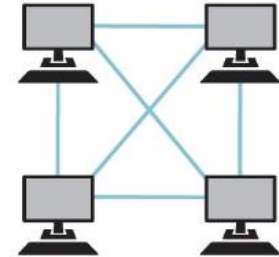


Las transiciones no son fáciles...

Lecciones aprendidas...



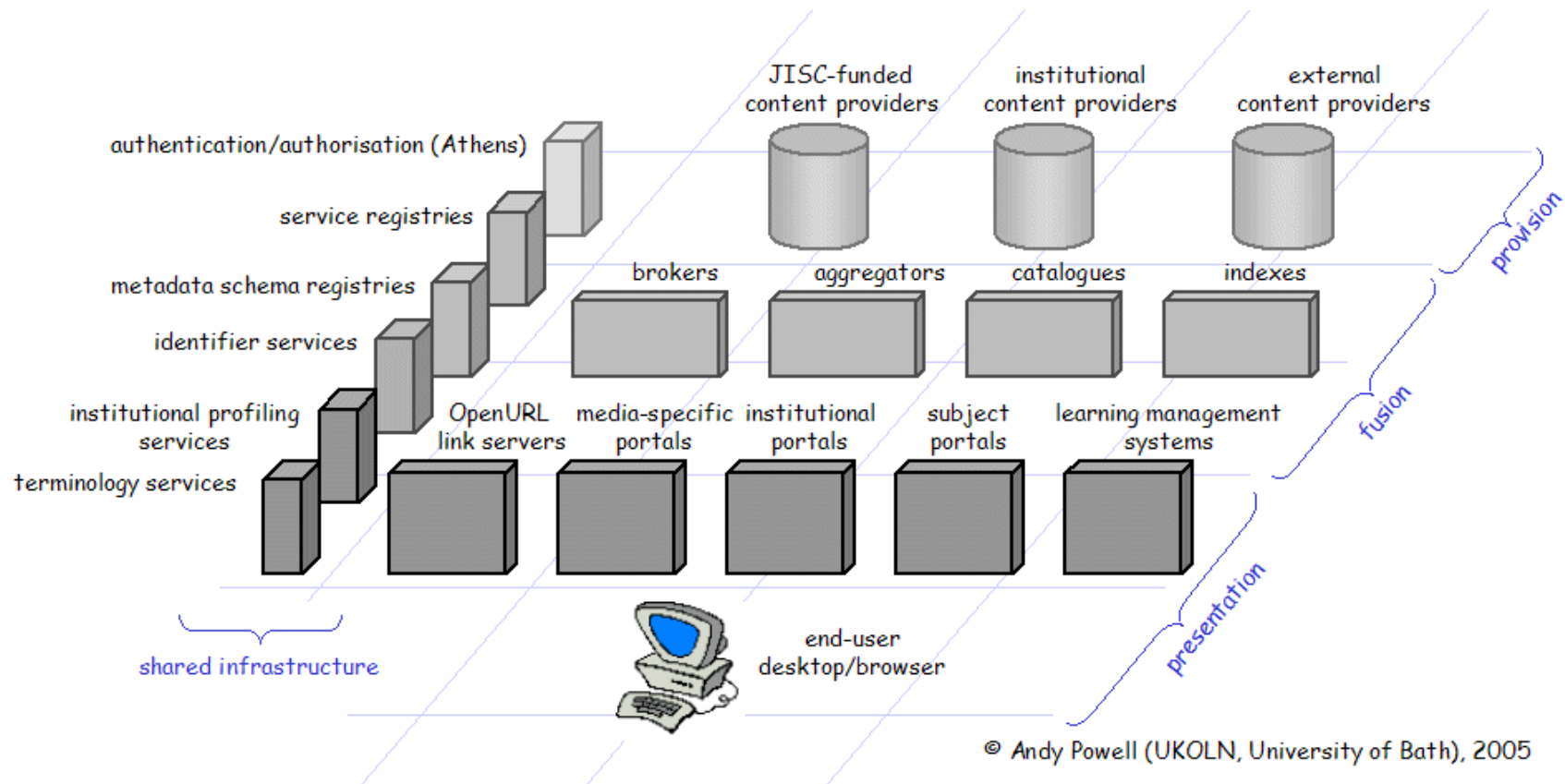
- ☐ Heterogeneity
- ☐ Openness
- ☐ Security
- ☐ Scalability
- ☐ Failure Handling
- ☐ Concurrency
- ☐ Transparency



Software/hardware extra en S.D.:
conseguir que la existencia de
múltiples elementos sea transparente

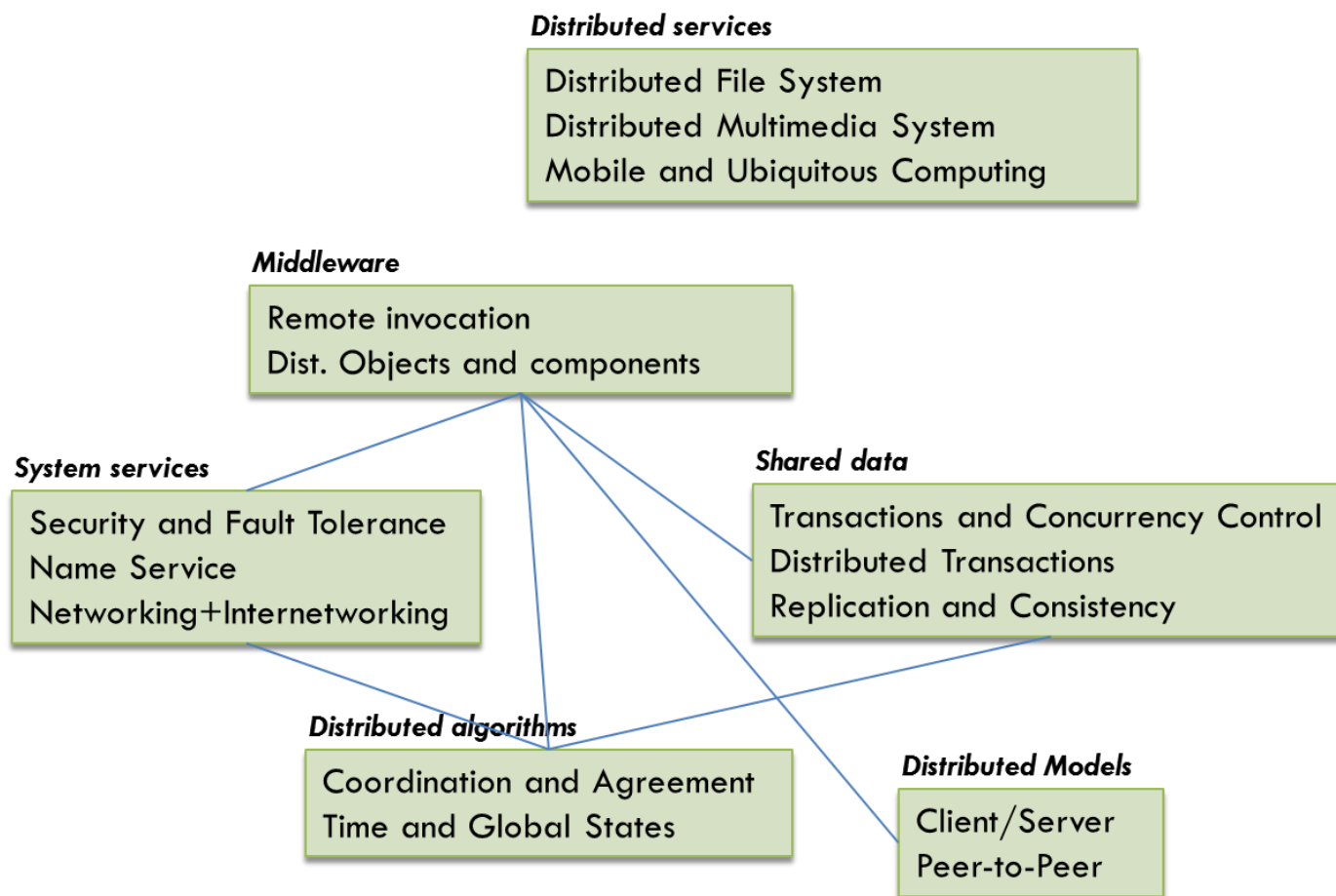
Las transiciones no son fáciles...

Lecciones aprendidas...



Las transiciones no son fáciles...

Lecciones aprendidas: elementos típicos...

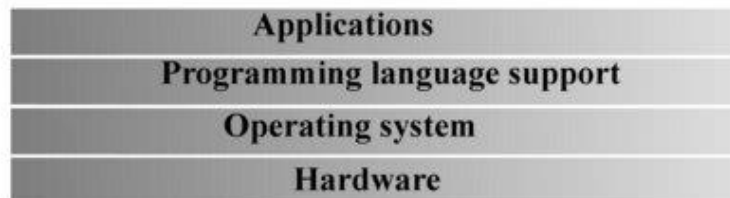


Las transiciones no son fáciles...

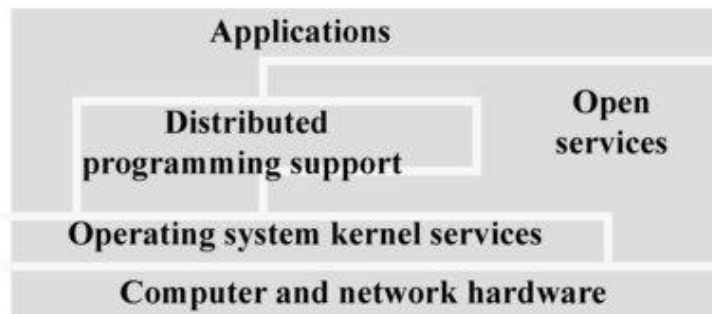
Lecciones aprendidas: arquitectura típica...

Distributed System Software Structure

Centralized
System



Distributed
System



Contenidos

Big
Big
Big
Big
Big

- Inicios
- Transición
- (re)Evolución

Sistemas distribuidos: mayor tamaño...

Virtually centralized ☺



Distributed

Distributed system:
existence of multiple elements is transparent



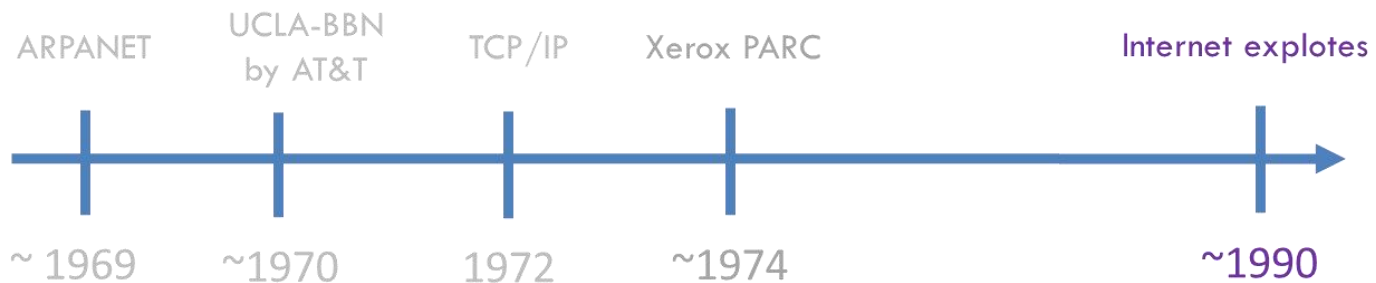
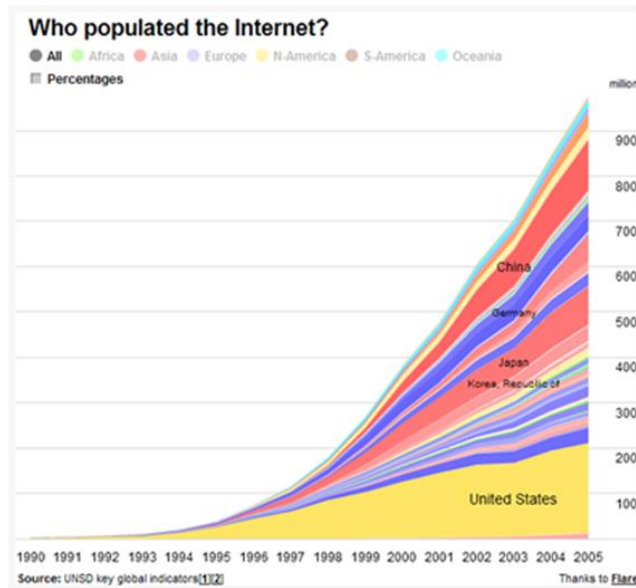
Cloud

Cloud computing:
Ubiquitous, on-demand access to
shared pool of computing resources

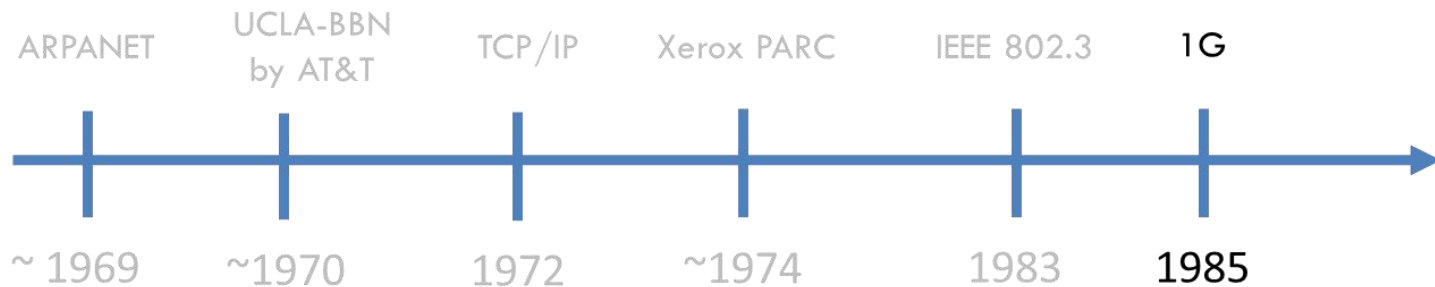
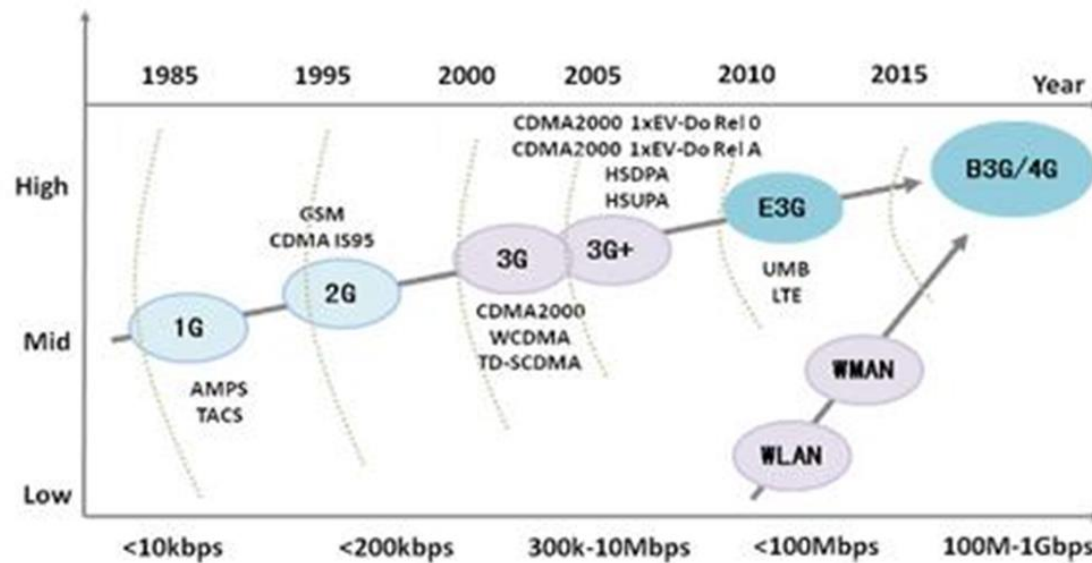


Sistemas distribuidos: mayor tamaño...

Todas las casas conectadas

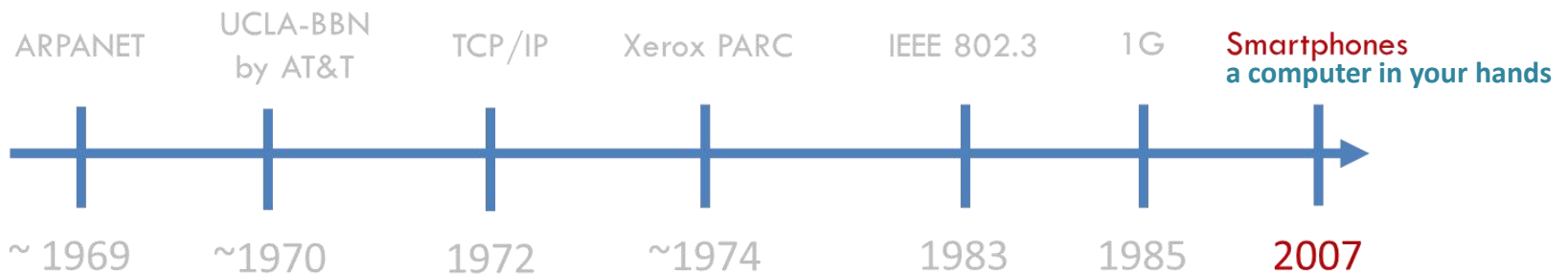
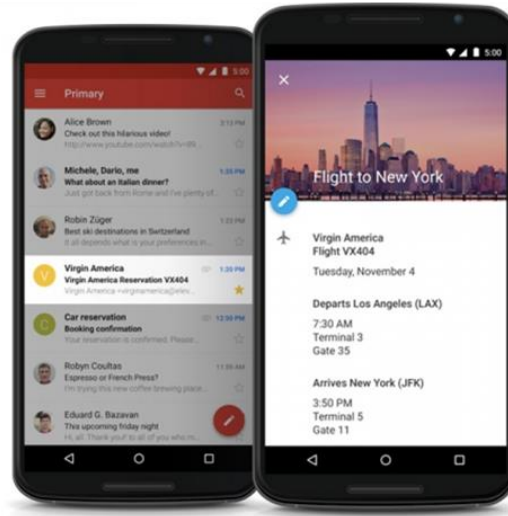
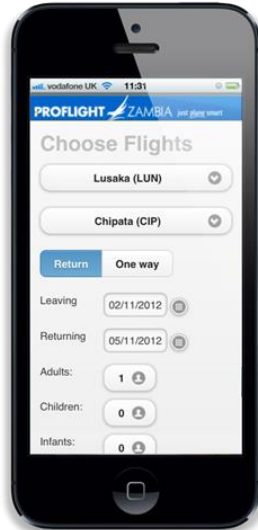


Sistemas distribuidos: movilidad...



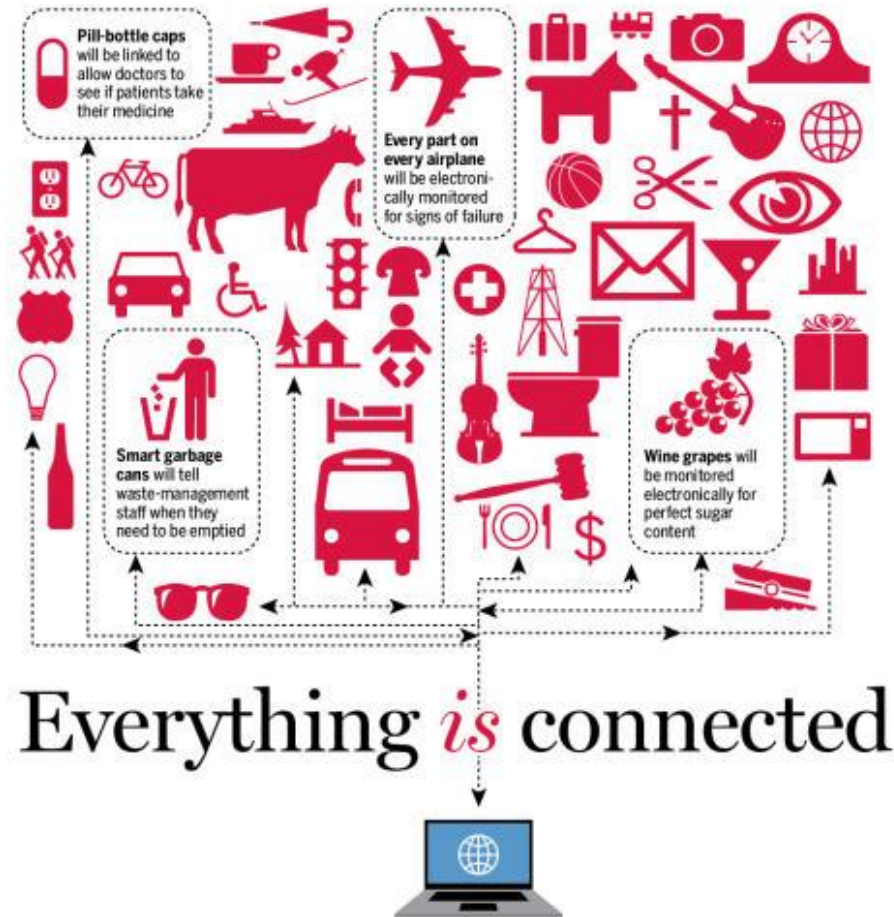
Sistemas distribuidos: mayor tamaño...

Todas las **personas** conectadas



Sistemas distribuidos de mayor tamaño...

Todas las cosas conectadas



PAU/BAY AREA NEWS GROUP

Ejemplo de IoT:

PI₀ + HTML5 Server-Sent Events

Feature

PI ZERO IN PICTURES

RASPBERRY PI ZERO



PI ZERO

IN PICTURES

We've been so enamoured with the size of the Raspberry Pi Zero that everyone here got into a small competition to show off their best size comparison photos!

Send us your best comparison shots on Twitter to @TheMagPi



RASPBERRY PI ZERO



Below: A LEGO Minifigure is just a bit taller than the Zero is wide. It's about one Minifigure wide and under two Minifigures long. Also, they can easily carry one!



Below: A classic, we're working on making a fully working Pi Zero - battery and everything - in an Altoids mini tin. You can fit six in one!



Below: Putting a Raspberry Pi Zero in a pack of cards sounds like proper James Bond territory, although you'd lose the cards you need to play poker.



RASPBERRY PI ZERO: IN NUMBERS

\$5 **3X**

The Raspberry Pi Zero is the first ever full computer that costs only \$5. But it's nearly three times smaller than the original Raspberry Pi.

143 **1080P**

The Raspberry Pi Zero can easily play 1080p video, even though it's so tiny and drives very little power.

2.5KG

It's so small, you could fit 143 Raspberry Pi Zeros over the screen of a 32" HD TV!

8128 MFLOPS

That's 100GB of RAM with 200 computing cores in a single box of Pi Zeros. The box only weighs 2.5kg.

512MB **160mA**

It also launches with twice as much RAM as the very first Raspberry Pi did (it 256MB).

Hooked up to a 1080p TV with a mouse and keyboard attached, the Pi Zero draws a tiny 160mA. The electricity bill will be the last thing on your mind!

10 | **MagPi** December 2015

raspberrypi.org/magpi

raspberrypi.org/magpi

December 2015 **MagPi** | 11

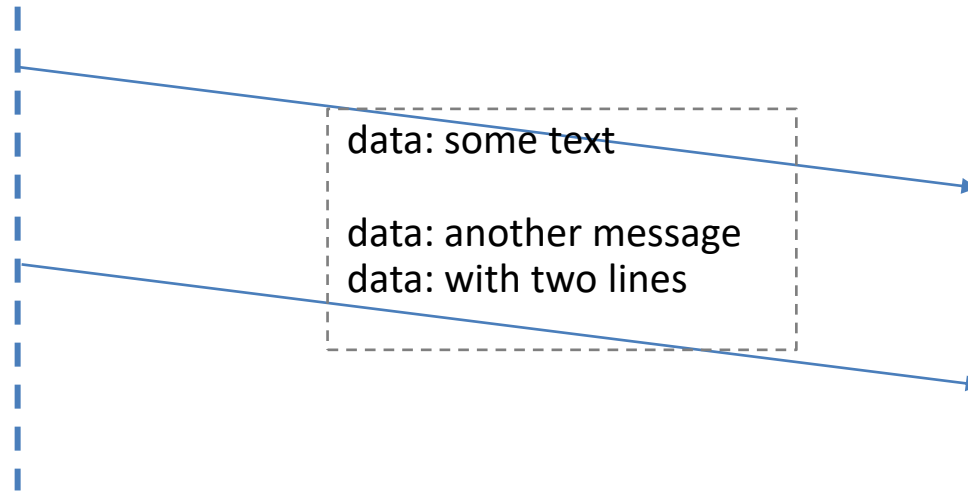
Ejemplo de IoT:

PI₀ + HTML5 Server-Sent Events



Server

Client



- Web page gets updates from a server
 - without ask for updates, the updates came automatically
- Examples: Twitter updates, stocks prices, news feeds, etc.

Ejemplo de IoT:

PI₀ + HTML5 Server-Sent Events



Server
(demo.php)

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');

while (1) {
    echo "data: {'time': " . date('r') . "}\\n\\n";
    ob_flush(); flush();
    sleep(1);
}
?>
```

Client
(demo.html)

```
<span id="result"></span>
<script>
var source = new EventSource("demo.php");
source.onmessage = function (e) {
    var ref = document.getElementById("result") ;
    ref.innerHTML += e.data + "<br>";
};
source.onerror = function(e) { alert("Problems..."); };
</script>
```

- Web page gets updates from a server
 - without ask for updates, the updates came automatically
- Examples: Twitter updates, stocks prices, news feeds, etc.

Ejemplo de IoT:

PI₀ + HTML5 Server-Sent Events



Server
(demo.sh)

```
echo "HTTP/1.1 200 OK"
echo "Access-Control-Allow-Origin: *"
echo "Content-Type: text/event-stream"
echo "Cache-Control: no-cache"
echo ""

while [ 1 ]; do
    T=$(date +%H:%M:%S)
    echo "data: {'timestamp': $T}\n\n"
    sleep 1
done
```

(demo-nc.sh)

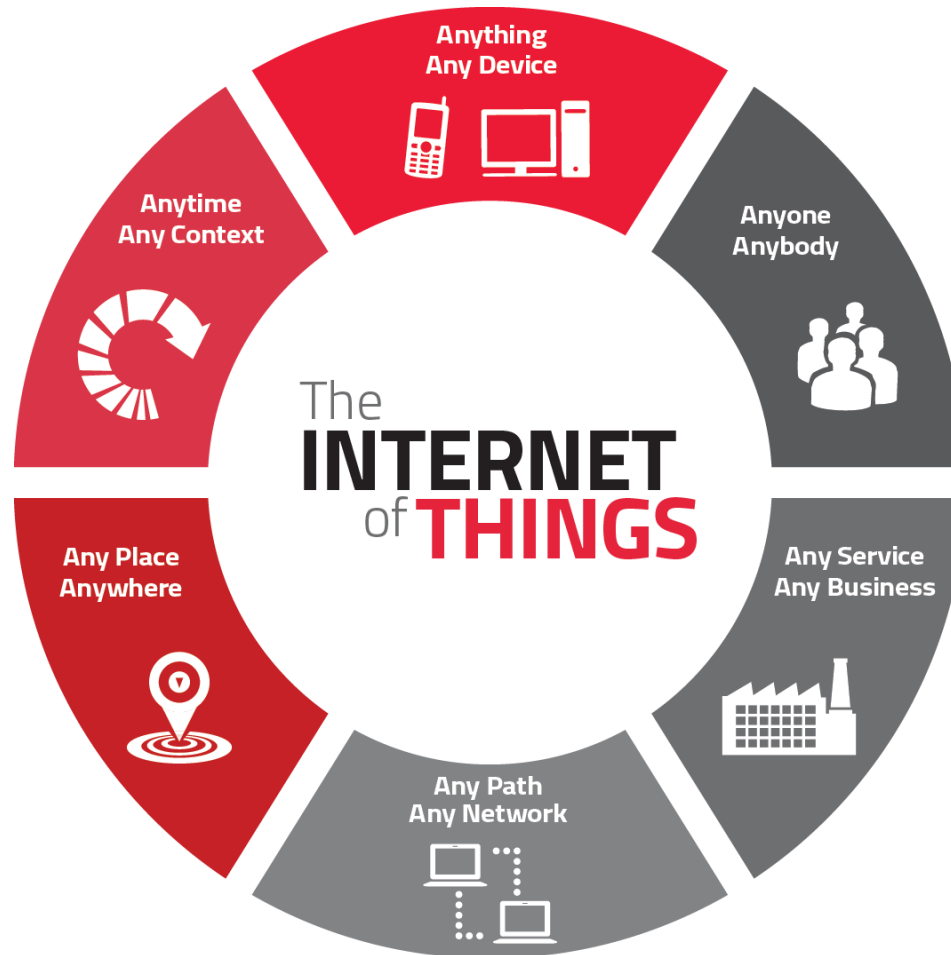
```
./demo.sh | nc -l -p 8080
```

Client
(demo.html)

```
<!DOCTYPE html>
<html>
<head> <meta charset="utf-8" /> </head>
<body>
  <script>
    var s = new EventSource('http://<ip>:8080');
    s.onmessage = function(e) {
      document.body.innerHTML += e.data + '<br>';
    };
  </script>
</body>
</html>
```

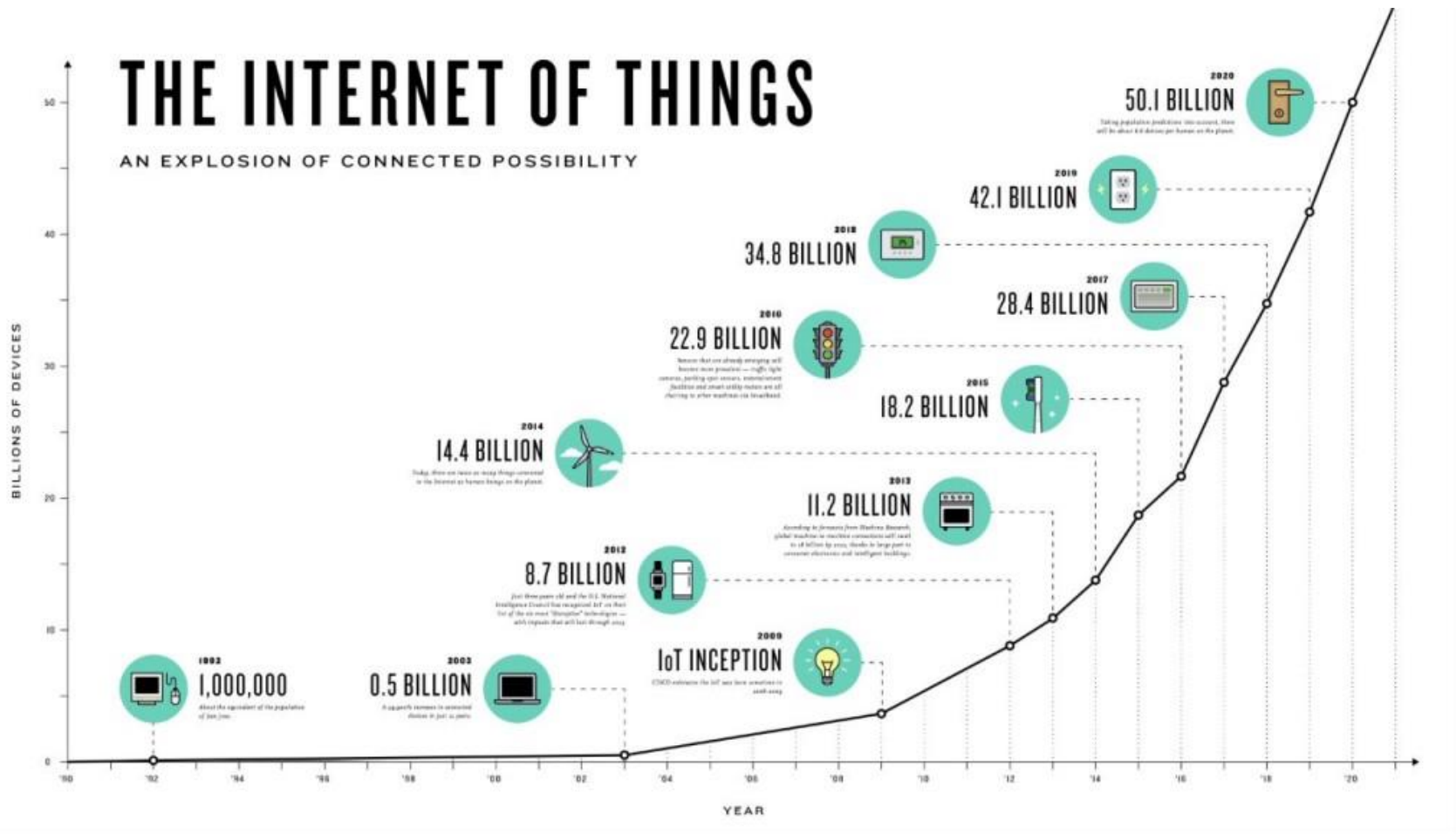
Sistemas distribuidos de mayor tamaño...

Internet of Things (IoT)



Sistemas distribuidos de mayor tamaño...

Internet of Things (IoT)



Contenidos

Big
Big
Big
Big
Big

- Inicio
- Transición
- (re)Evolución
- (re)Retos

Dependencia cada vez mayor...

Virtually centralized ☺



Distributed

Distributed system:
existence of multiple elements is transparent



Cloud

Cloud computing:
Ubiquitous, on-demand access to
shared pool of computing resources

https://www.dcsorg.com/images/image_centralized_management.jpg

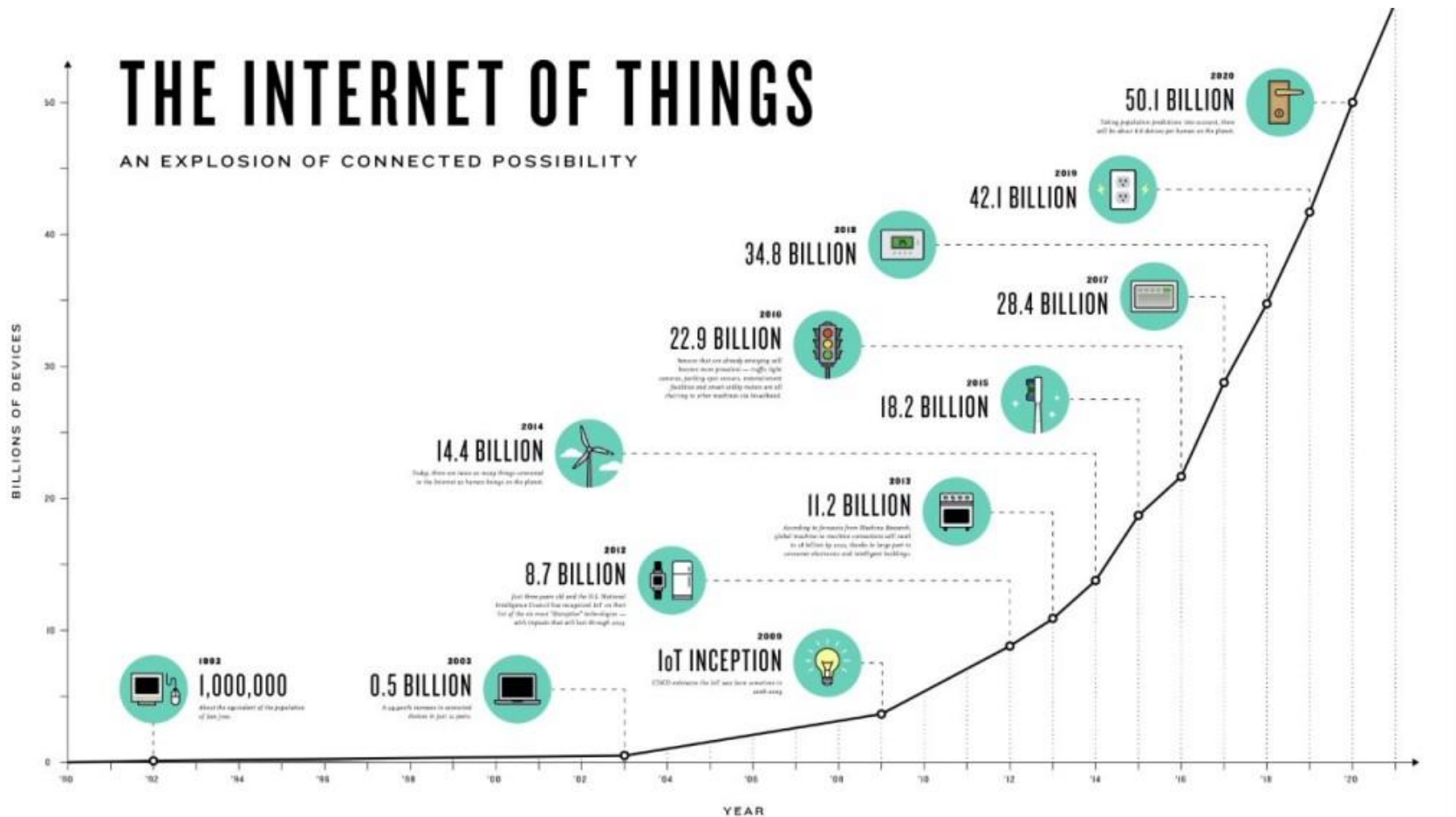
<http://www.hsi.es/images/cc.png>

Dependencia cada vez mayor...

criticidad con difícil vuelta atrás



Nuevos retos...



Big data is coming...

From a Molehill To a Mountain

A timeline of digital data

Everyone knows that the amount of digital data has exploded in recent years. But it isn't always easy to put today's mountain of data in perspective.

To illustrate the data boom, here's a timeline with snapshots of some of the largest corporate data collections from each decade since the Univac computer was the hot new thing.

—Matthew Kassel

2010s | 100 Petabytes

Facebook Is Facebook's data hoard bigger than Google's today? Some say yes, some say no. According to Facebook, its user content makes up more than 100 petabytes of stored photos and video. And analyzing that data generates about 500 terabytes of new information every day—more than 2½ times the size of a '90s Wal-Mart data cache.

2000s | 25 Petabytes

Google The explosion of the Internet in the '90s set the stage for Web-based companies to emerge in the following decade as the global leaders in big data.

1990s | 180 Terabytes

Wal Mart Wal-Mart in this decade became the largest American bricks-and-mortar retail operation, and, it is believed, had the biggest commercial data warehouse in the world. To put this number in perspective, experts have estimated that Amazon.com in the late '90s had single-digit terabytes of stored data.

1980s | 450 Gigabytes

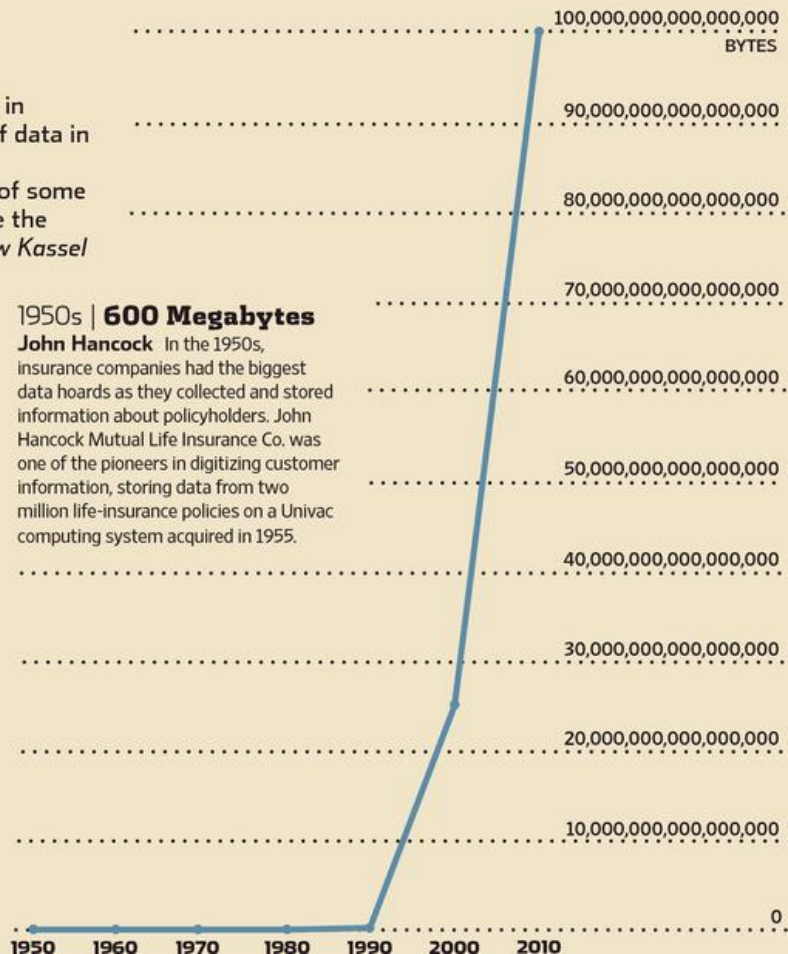
CitiCorp's NAIB In the 1980s, banks were at the forefront of data growth, with ATMs coming into vogue and banks focusing on collecting and analyzing transaction data for all their businesses.

1970s | 80 Gigabytes

Federal Express Cosmos FedEx's Cosmos system, introduced in the 1970s, allowed the company to scan and track its huge volume of packages being shipped around the world.

1960s | 807 Megabytes

American Airlines Sabre In the 1960s, American Airlines developed Sabre, a flight-reservation system built around one of the largest IBM computing systems available. Sabre was one of the first online computing systems, allowing the airline to keep track of an immense matrix of reservations, flight schedules and seat inventories.



Big data is coming...

Ciencia
Industria
Logs de sistema
...

From a Molehill To a Mountain

A timeline of digital data

Everyone knows that the amount of digital data has exploded in recent years. But it isn't always easy to put today's mountain of data in perspective.

To illustrate the data boom, here's a timeline with snapshots of some of the largest corporate data collections from each decade since the Univac computer was the hot new thing.

—Matthew Kassel

2010s | 100 Petabytes

Facebook Is Facebook's data hoard bigger than Google's today? Some say yes, some say no. According to Facebook, its user content makes up more than 100 petabytes of stored photos and video. And analyzing that data generates about 500 terabytes of new information every day—more than 2½ times the size of a '90s Wal-Mart data cache.

2000s | 25 Petabytes

Google The explosion of the Internet in the '90s set the stage for Web-based companies to emerge in the following decade as the global leaders in big data.

1990s | 180 Terabytes

Wal Mart Wal-Mart in this decade became the largest American bricks-and-mortar retail operation, and, it is believed, had the biggest commercial data warehouse in the world. To put this number in perspective, experts have estimated that Amazon.com in the late '90s had single-digit terabytes of stored data.

1980s | 450 Gigabytes

CitiCorp's NAIB In the 1980s, banks were at the forefront of data growth, with ATMs coming into vogue and banks focusing on collecting and analyzing transaction data for all their businesses.

1970s | 80 Gigabytes

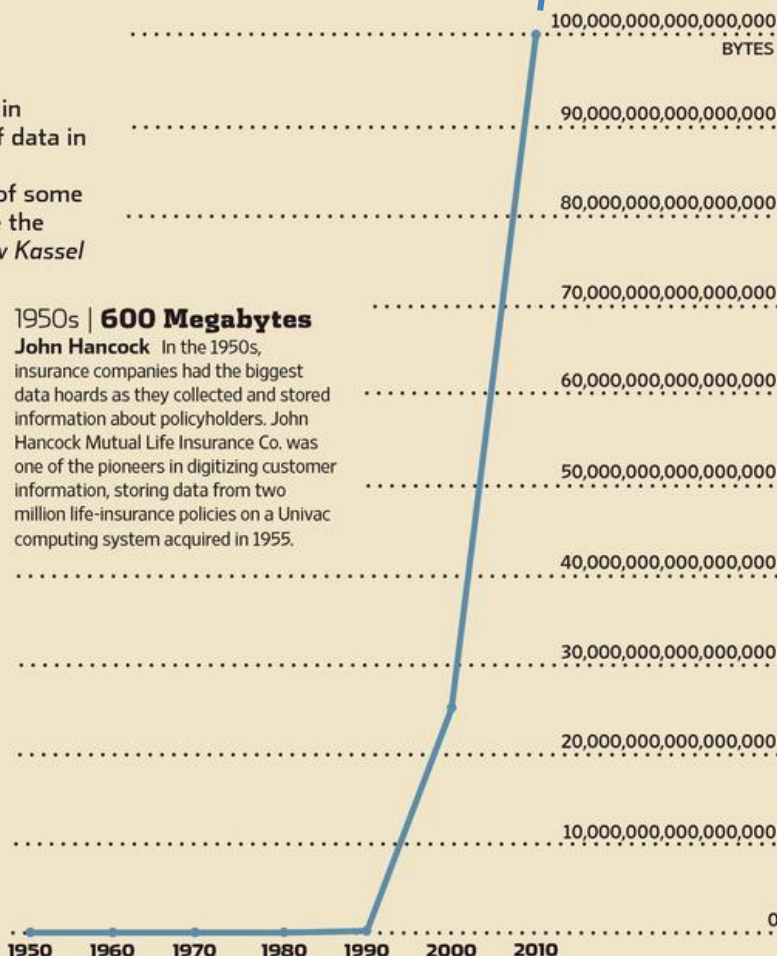
Federal Express Cosmos FedEx's Cosmos system, introduced in the 1970s, allowed the company to scan and track its huge volume of packages being shipped around the world.

1960s | 807 Megabytes

American Airlines Sabre In the 1960s, American Airlines developed Sabre, a flight-reservation system built around one of the largest IBM computing systems available. Sabre was one of the first online computing systems, allowing the airline to keep track of an immense matrix of reservations, flight schedules and seat inventories.

1950s | 600 Megabytes

John Hancock In the 1950s, insurance companies had the biggest data hoards as they collected and stored information about policyholders. John Hancock Mutual Life Insurance Co. was one of the pioneers in digitizing customer information, storing data from two million life-insurance policies on a Univac computing system acquired in 1955.



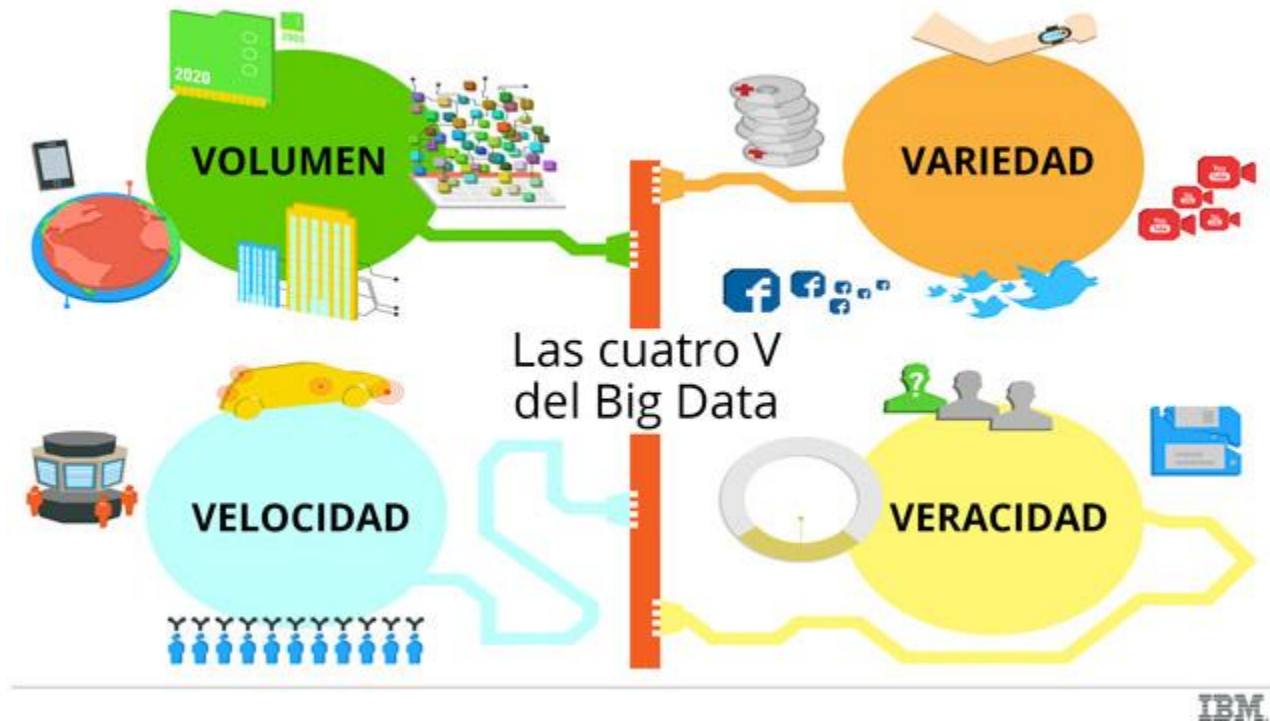
Big Data frequently used...



- Text mining
- Index building
- Graph creation and analysis
- Pattern recognition
- Prediction model
- ...



Big Data & Big Processing...



Las transiciones siguen sin ser fáciles...

Áreas de trabajo...

- Heterogeneity
 - Openness
 - Security
 - Scalability →
 - Failure Handling
 - Concurrency
 - Transparency
- A system is **scalable** if will remain **effective** when there is a significant increase in the number of resources and the number of users (represented by client programs).

Big Distributed Systems are coming...

□ Heterogeneity

□ Openness

□ Security

□ Scalability

□ Failure Handling

□ Concurrency

□ Transparency

□ A system is **scalable** if will remain **effective** when there is a significant increase in the number of resources and the number of users (represented by client programs).

□ Many challenges:

▣ Control of performance degradation

▣ Cost of physical resources

■ But preventing running out of resources

■ But avoiding performance bottlenecks

Contenidos

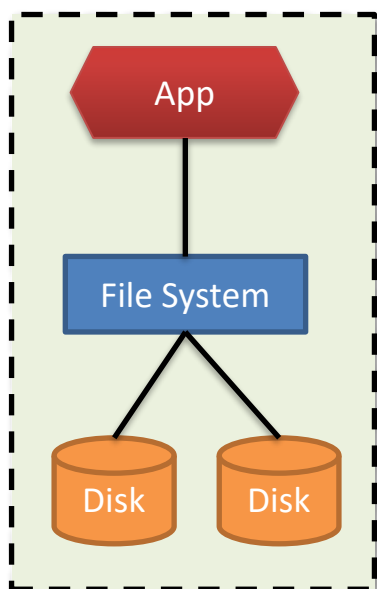
Big
Big
Big
Big
Big



- Inicio
 - Transición
 - (re)Evolución
 - (re)Retos
- **Inicio**
 - Hadoop
 - Ecosistema
 - Evolución

Repaso de opciones... (1/3)

DAS
Direct Attached Storage

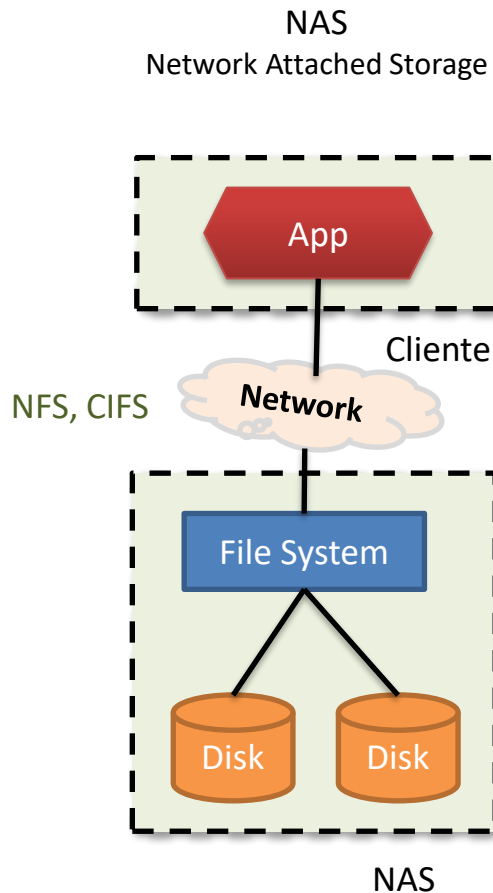


Ordenador



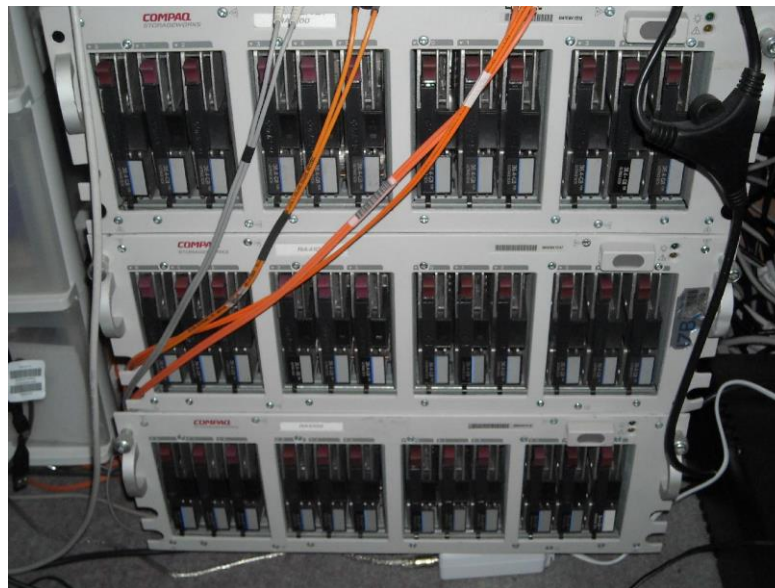
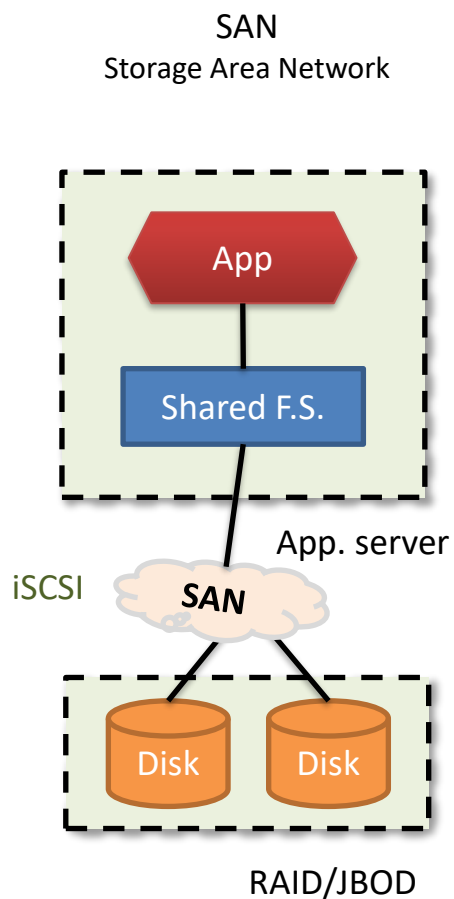
- V/I:
 - Rapidez, simplicidad
 - No **compartición**, **crecimiento** limitado

Repaso de opciones... (2/3)



- V/I:
 - Compartición (RO), crecimiento
 - Red limita velocidad y crecimiento

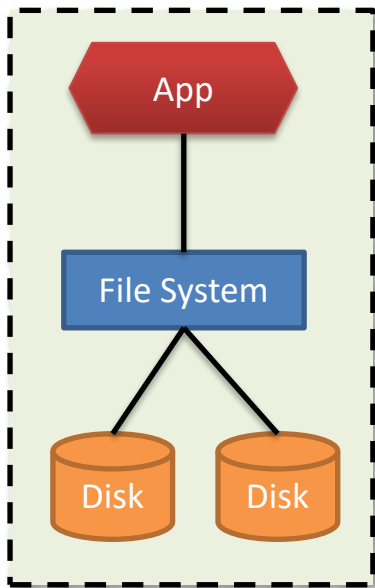
Repaso de opciones... (3/3)



- V/I:
 - Mejor crecimiento-velocidad
 - Complejidad, **coste**

Combinación de opciones...

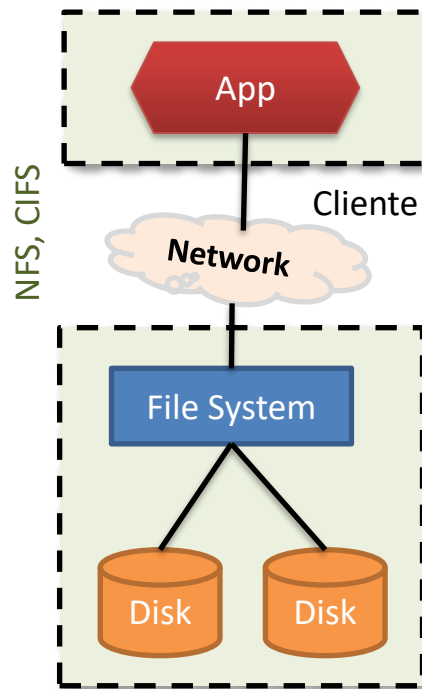
DAS
Direct Attached Storage



Ordenador



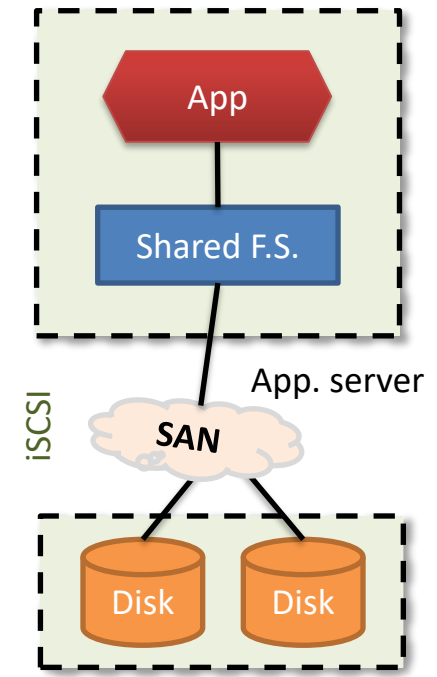
NAS
Network Attached Storage



NAS

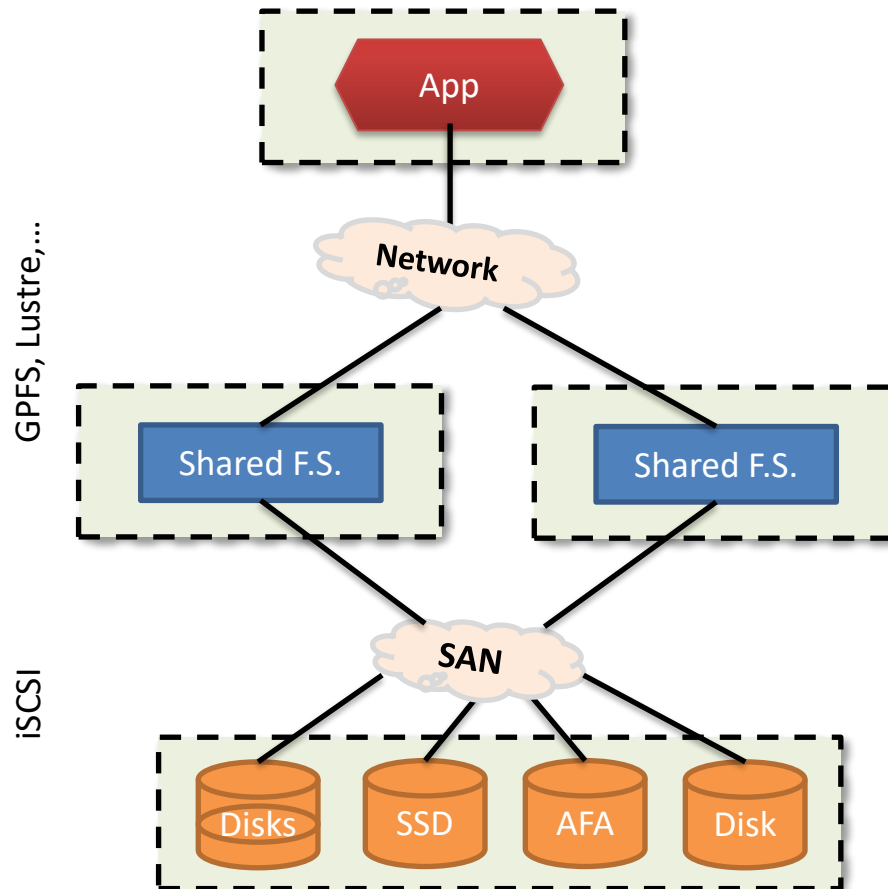


SAN
Storage Area Network



RAID/JBOD

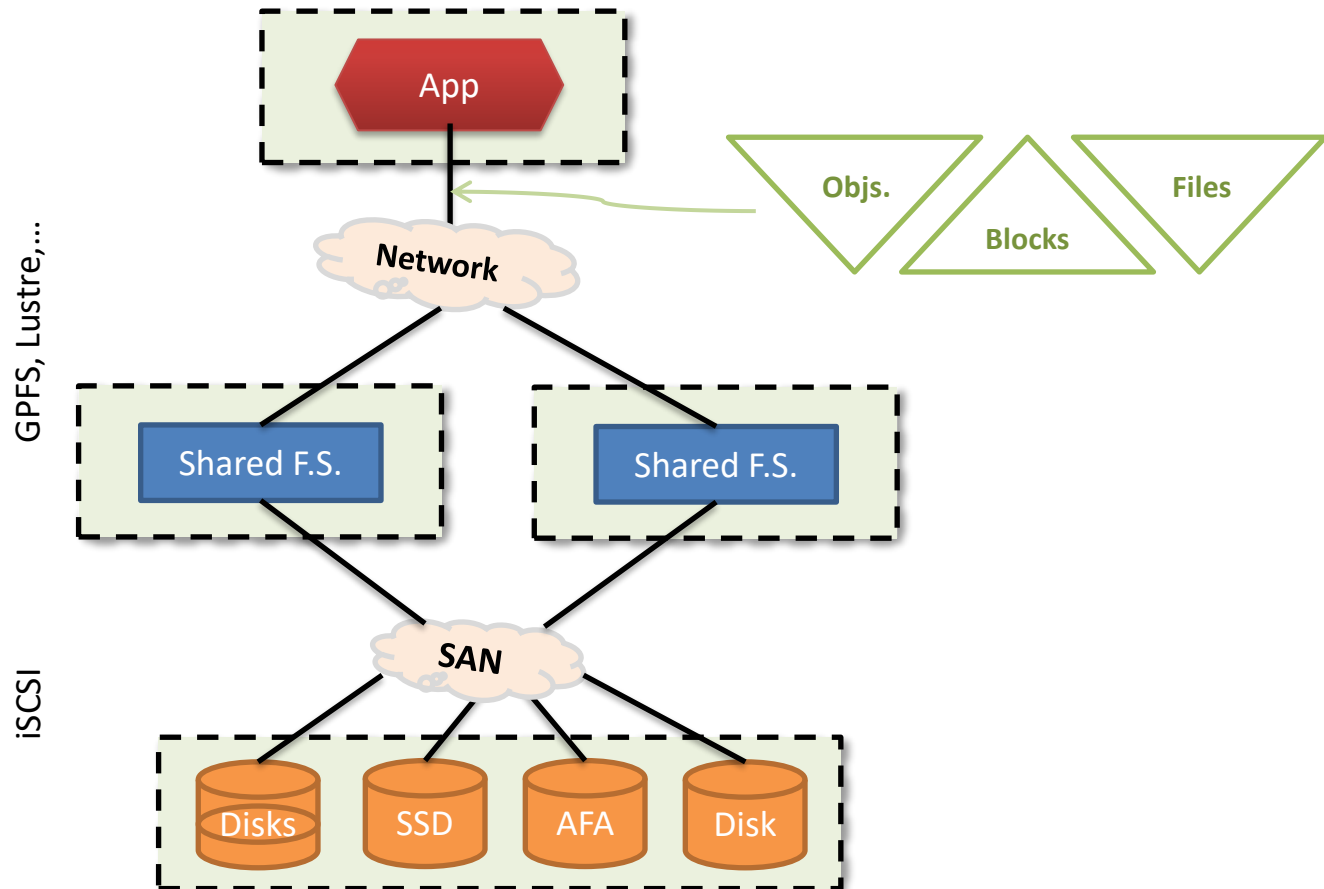
Combinación de opciones...



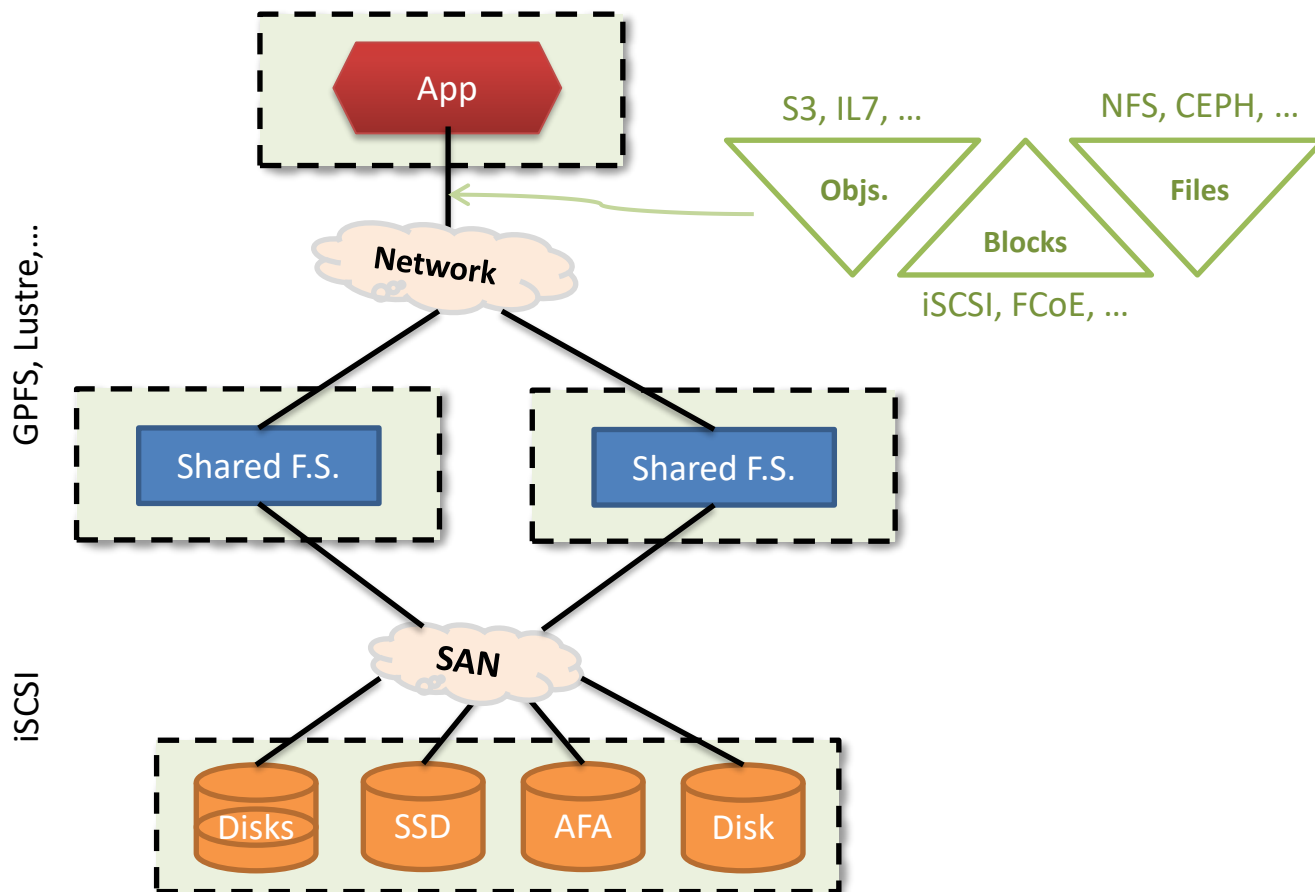
NAS + SAN



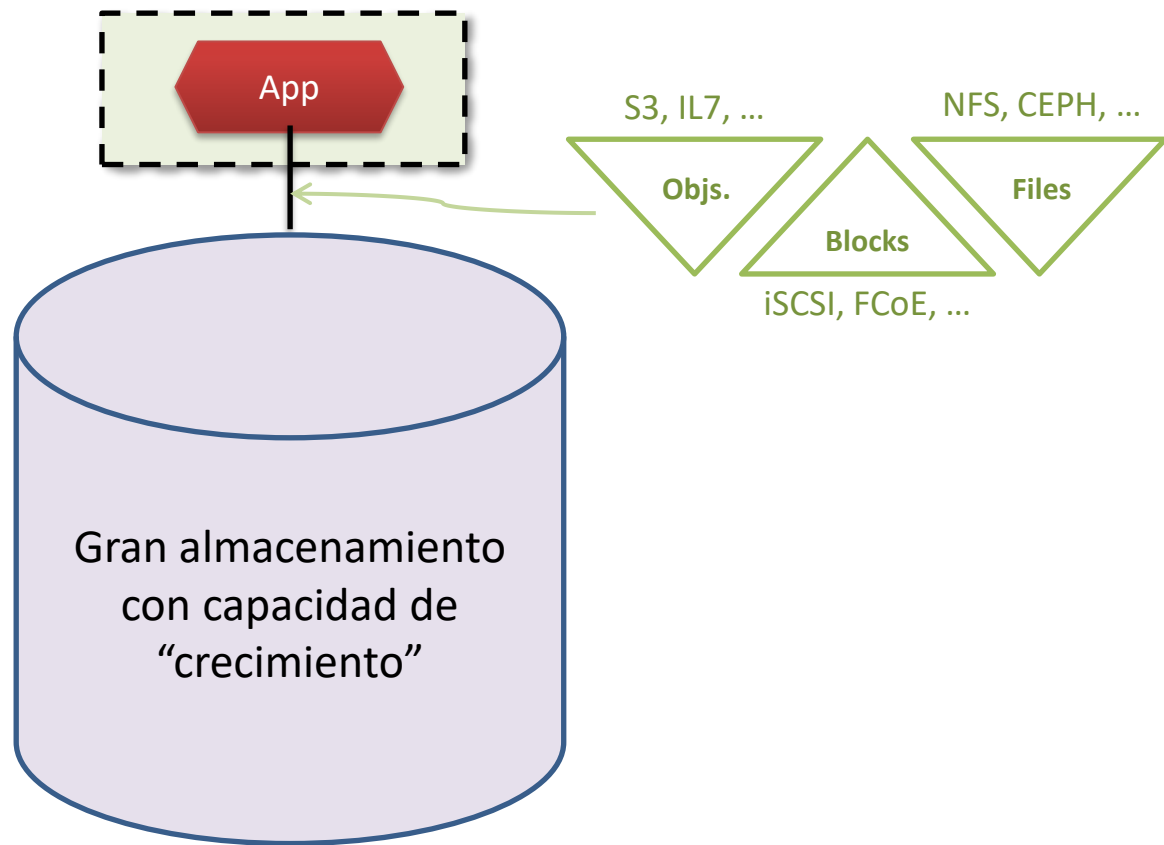
Combinación de opciones...



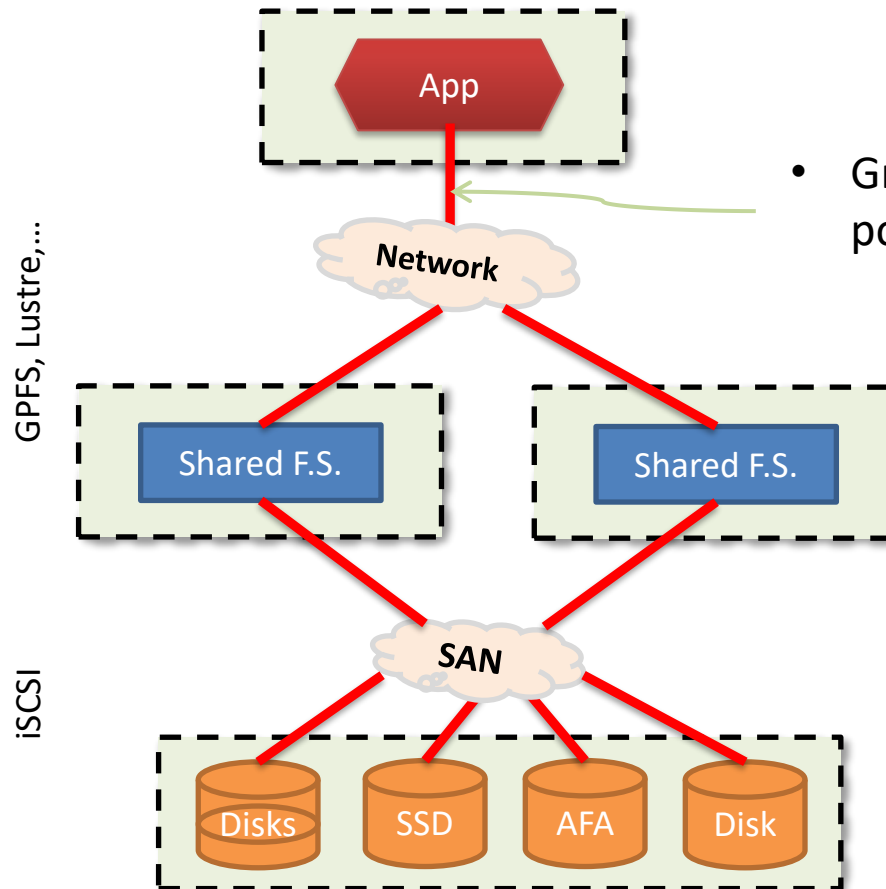
Combinación de opciones...



Combinación de opciones...

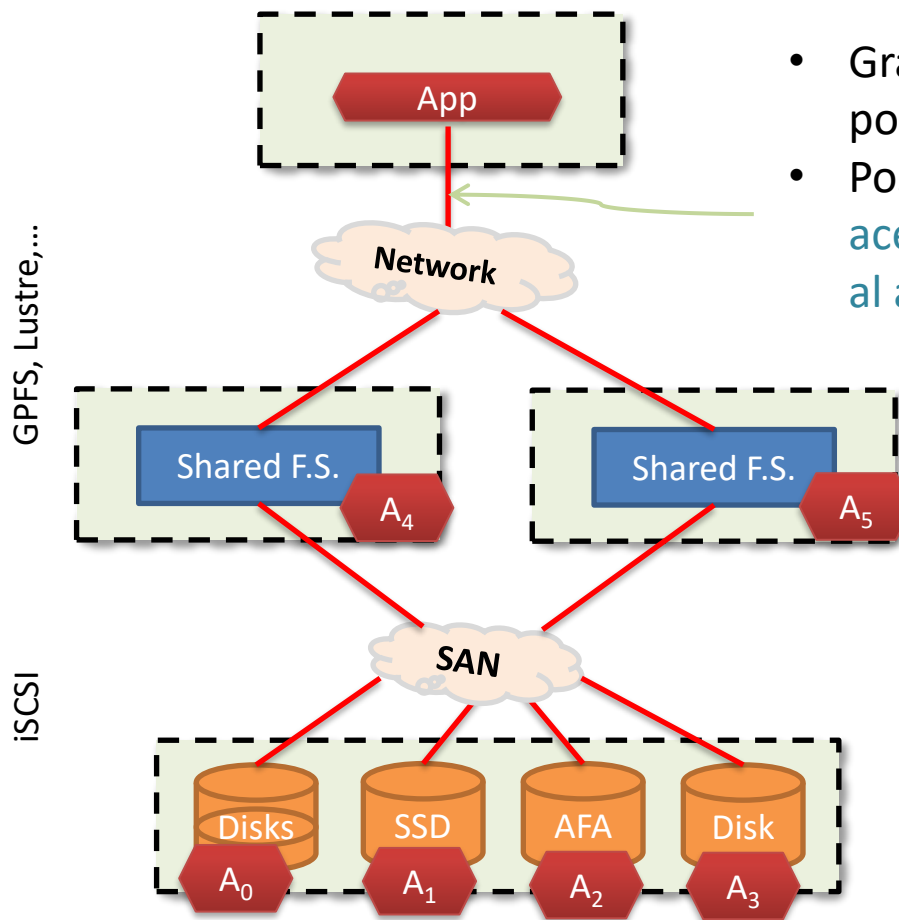


Problemas...



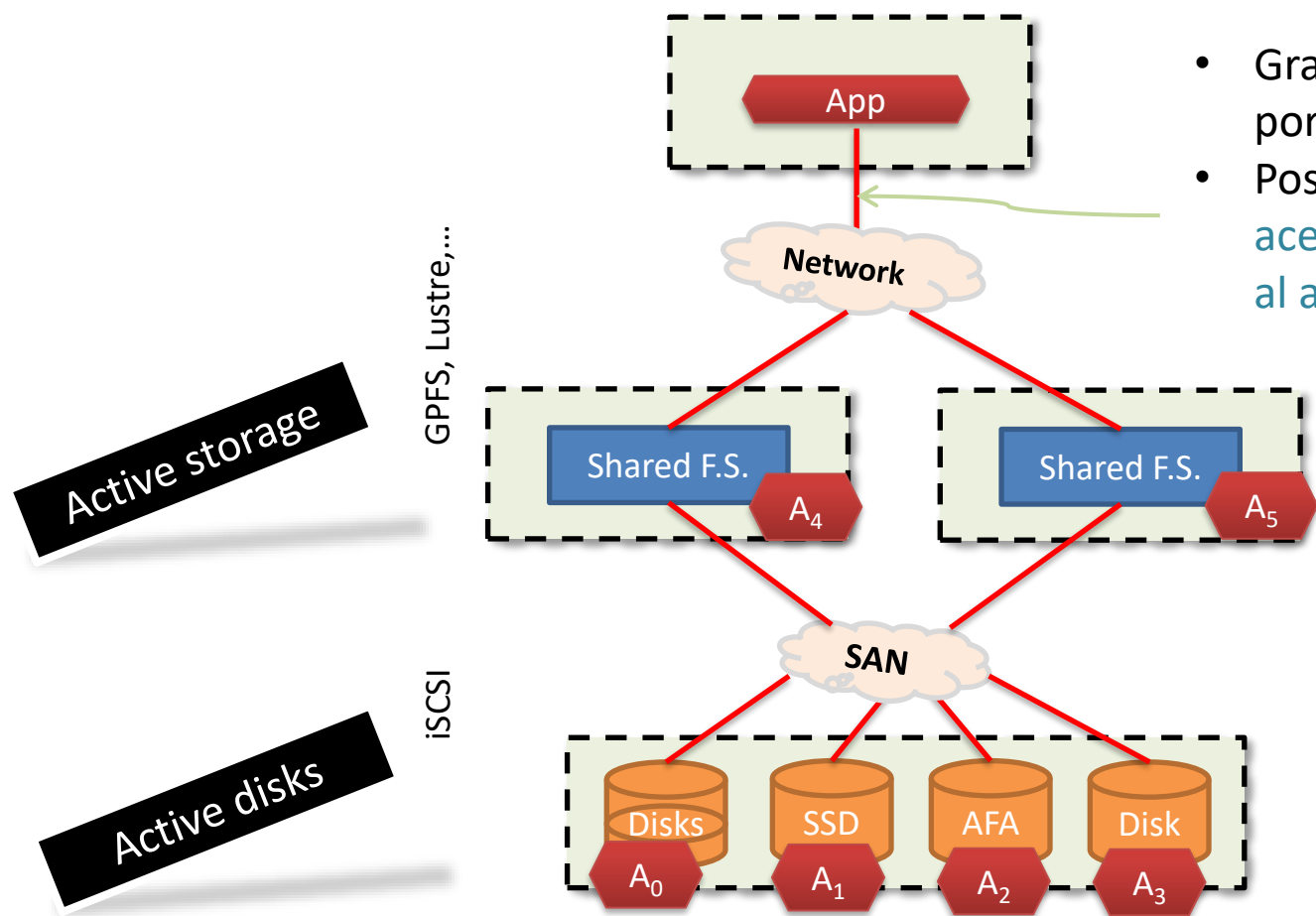
- Gran cantidad de tráfico por movimiento de datos.

Opciones para almacenamiento...



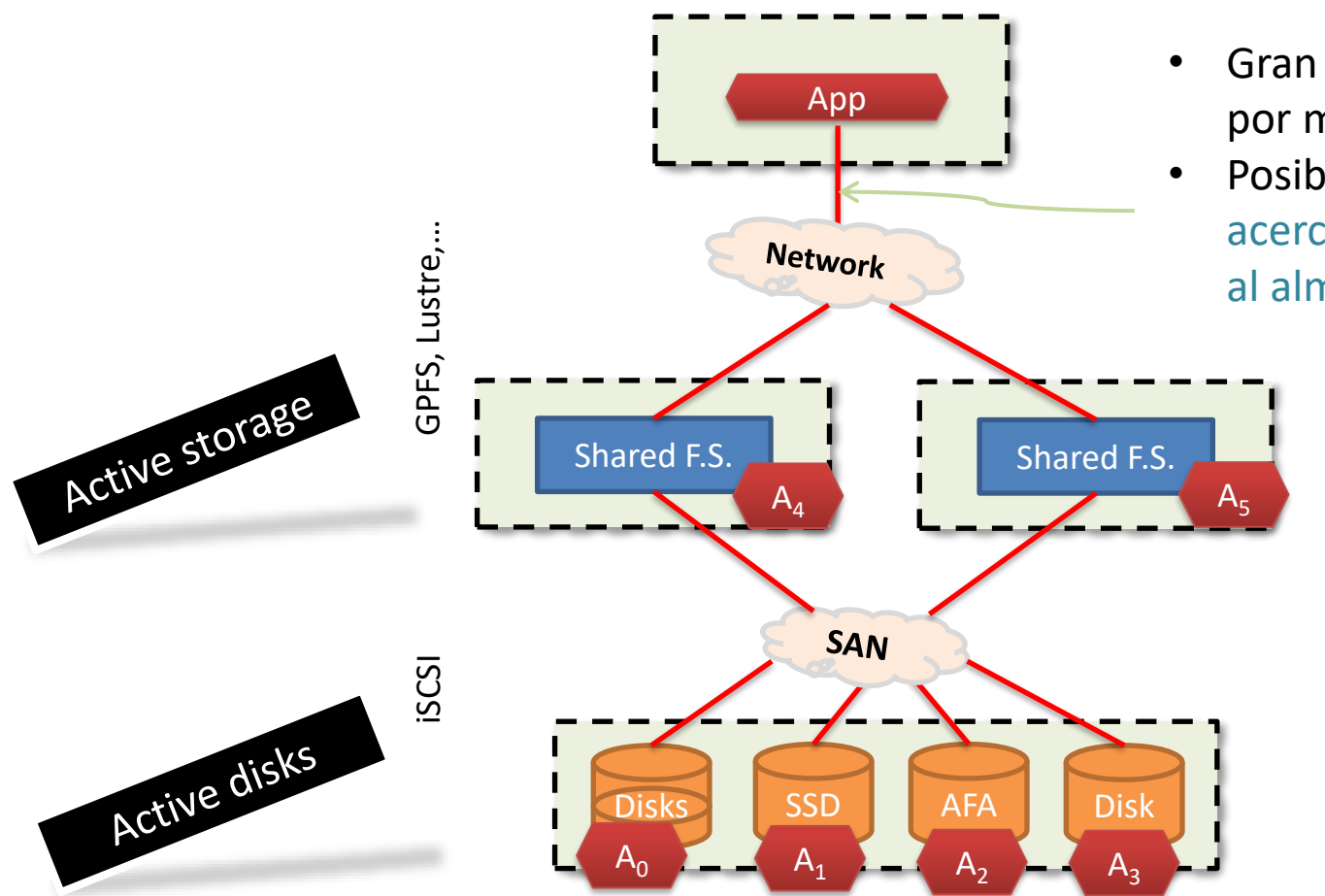
- Gran cantidad de tráfico por movimiento de datos.
- Posible disminuir si se acerca parte del cómputo al almacenamiento.

Opciones para almacenamiento...



- Gran cantidad de tráfico por movimiento de datos.
- Posible disminuir si se acerca parte del cómputo al almacenamiento.

Opciones para almacenamiento...



- Gran cantidad de tráfico por movimiento de datos.
- Posible disminuir si se acerca parte del cómputo al almacenamiento.

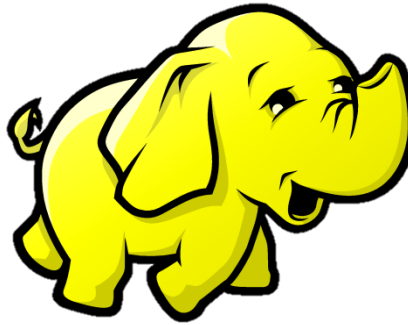


Buscar/crear la herramienta adecuada...



Contenidos

Big
Big
Big
Big
Big

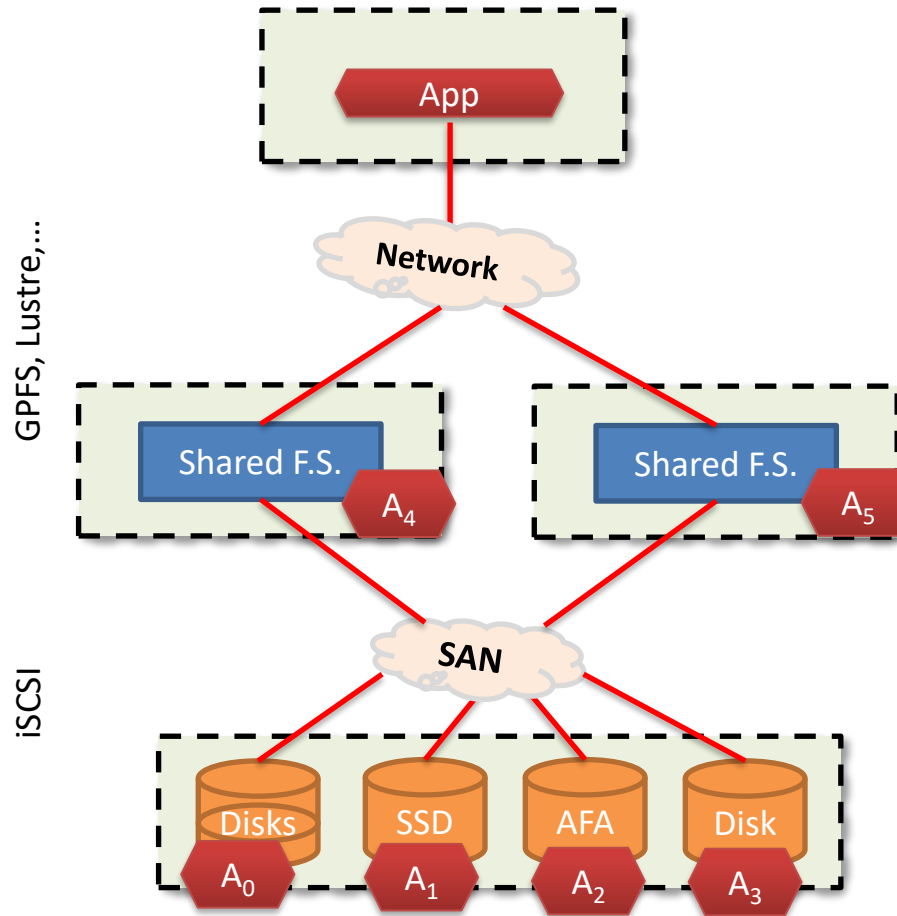


- Inicio
- Transición
- (re)Evolución
- (re)Retos
- Inicio
- **Hadoop**
- Ecosistema
- Evolución

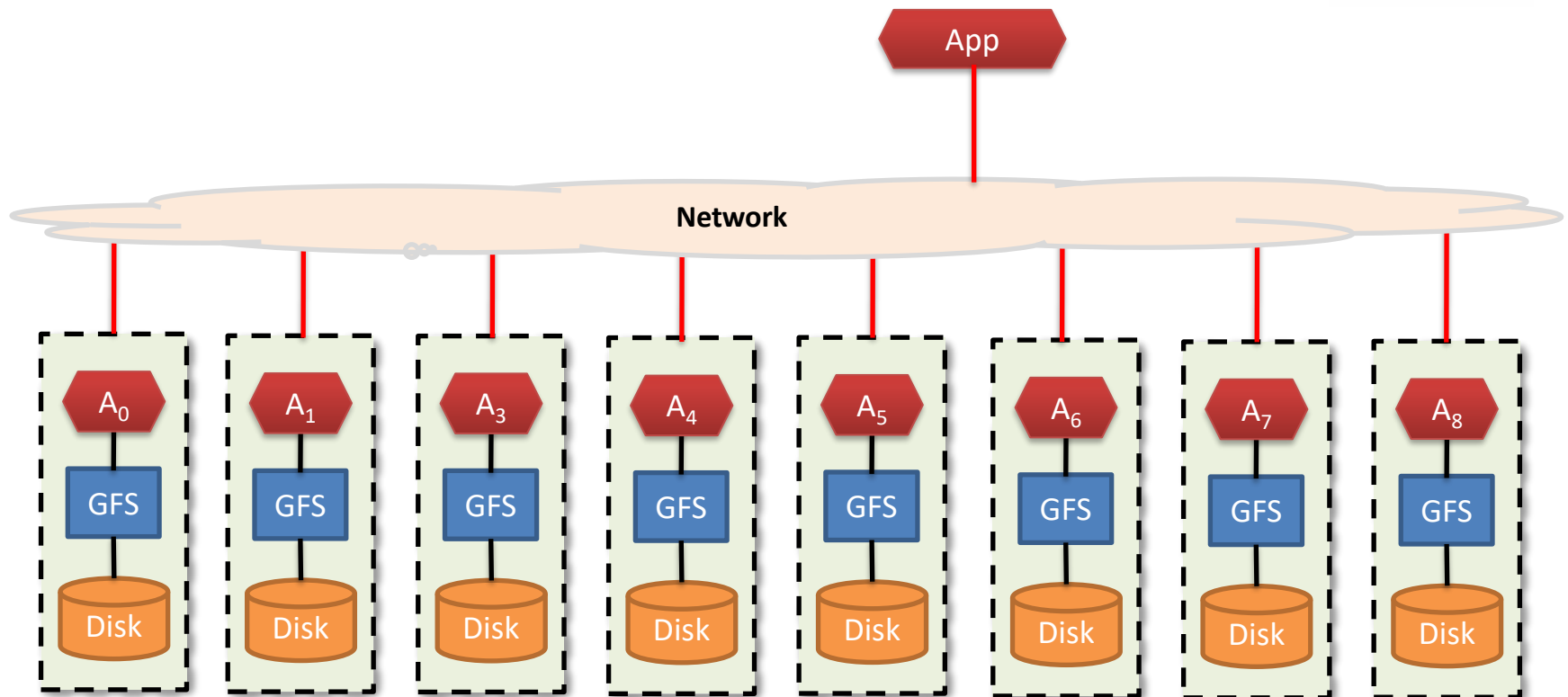
The Google File System

- Google presenta **MapReduce** y **Google File System (GFS)**
 - **MapReduce** es una propuesta para aplicar la misma función a particiones de datos (map) y luego se tiene el resultado procesando los resultados parciales (reduce)
 - **GFS** es una propuesta para almacenar petabytes de datos en muchas máquinas comunes, tratando con fallos, distribución, etc.

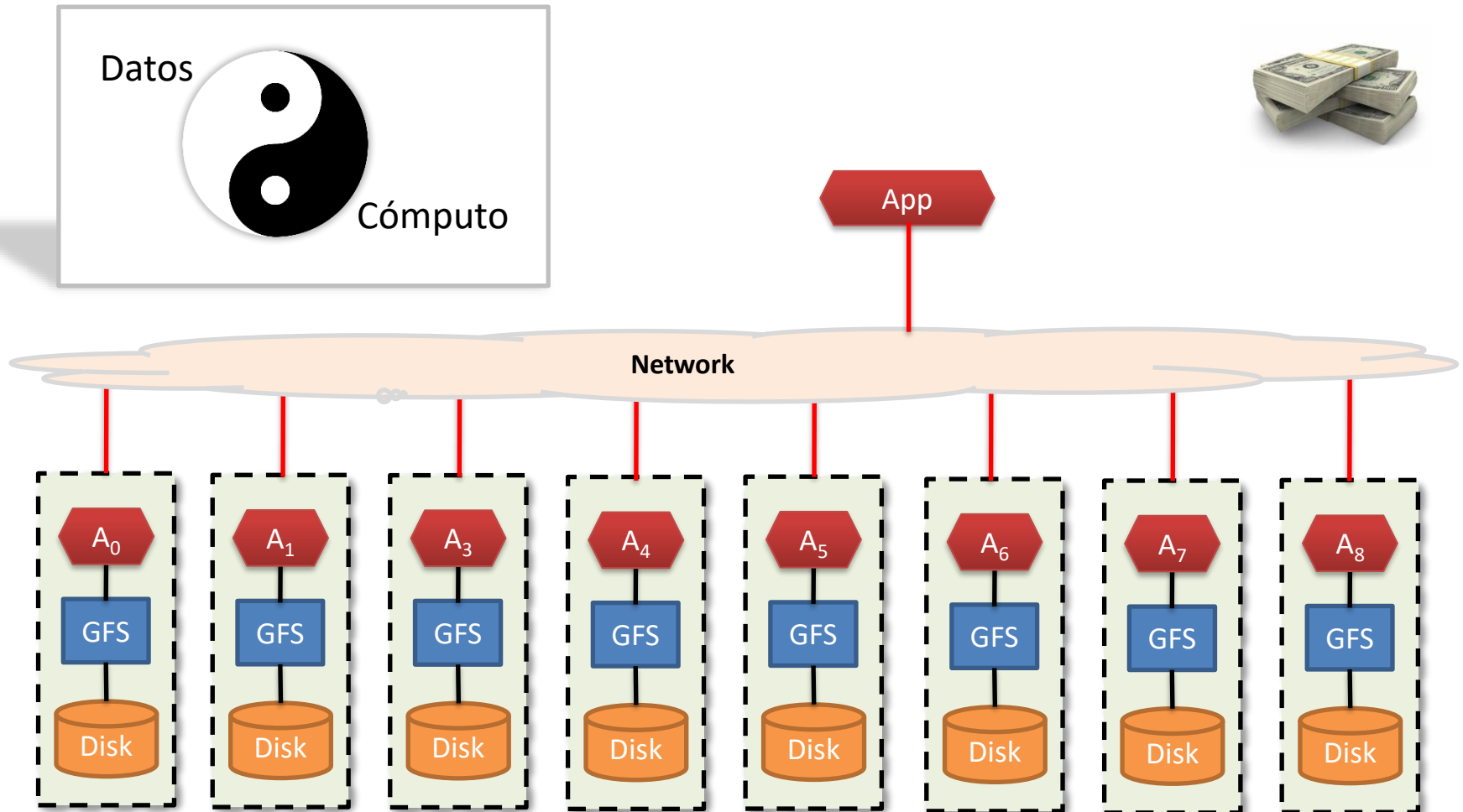
De esta opción...

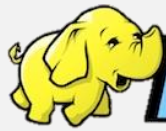


...a esta opción



...a esta opción

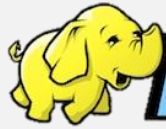




Breve historia

- **Doug Cutting** trabajando en Yahoo! e inspirado por estas tecnologías, inicia el desarrollo de **Hadoop**.
 - El proyecto usa Java, implanta las ideas detrás de GFS y MapReduce.
 - El logotipo se basa en el elefante amarillo que era el juguete favorito de su hijo.
 - Doug Cutting pasó a trabajar a **cloudera**.





Breve historia

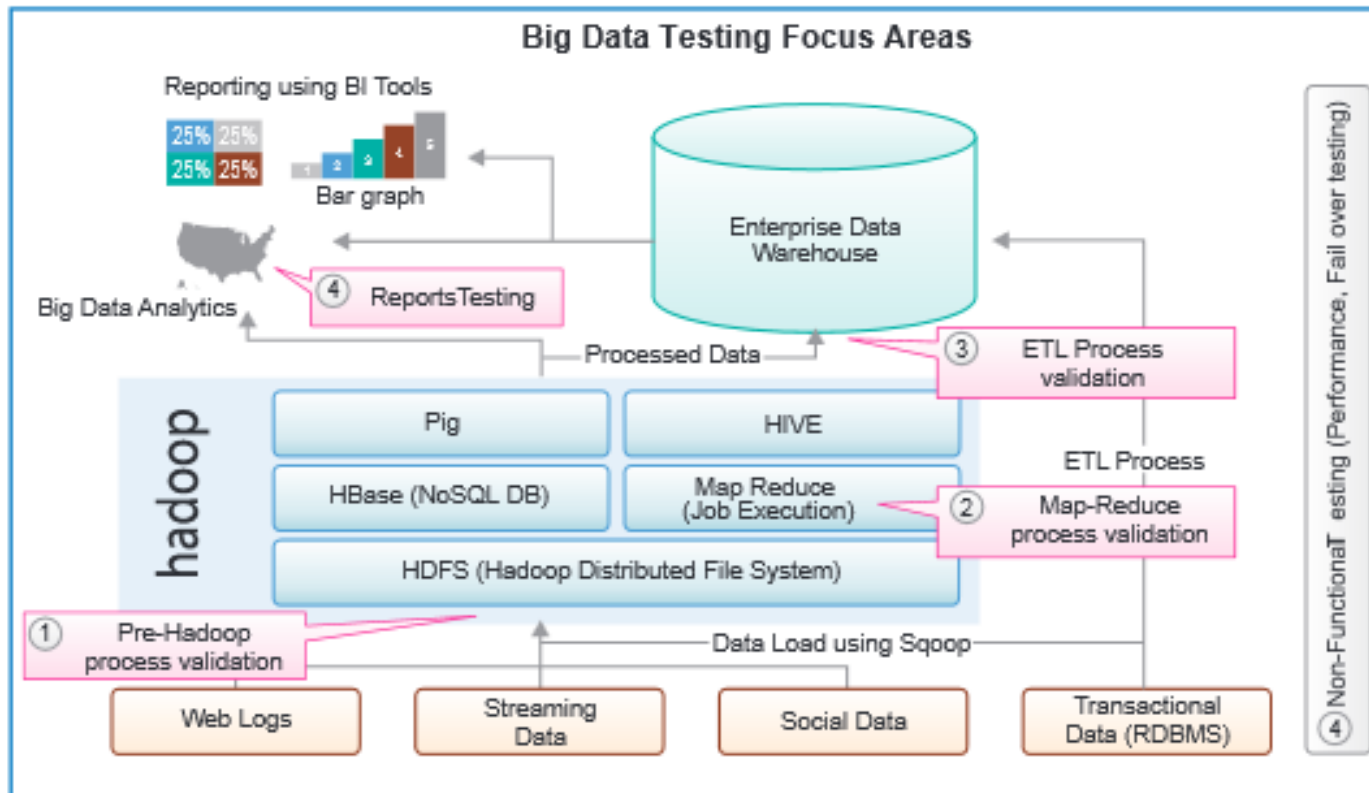
- Actualmente es un proyecto de **código abierto** bajo **licencia Apache**.
 - Su licencia ha facilitado que sea adoptado por un importante número de empresas.
- El **apoyo de IBM, Oracle, EMC, etc.** ha acelerado su implantación y su mejora en prestaciones.
 - Gran uso en proyectos tipo *big data*.
 - JPMorgan Chase: “We’re hiring, and we’re paying 10% more than the other guys.”

<http://hadoop.apache.org>

- [Apache Hadoop](#) es un proyecto software *open source* para computación distribuida escalable y de confianza (*reliable*).
- Ofrece un framework que [permite la computación distribuida de grandes conjuntos de datos mediante clusters de ordenadores usando modelos de programación simples](#).
 - Diseñado para poder pasar de un solo servidor a miles de máquinas (scale-up), donde cada una ofrece tanto computación como almacenamiento.
 - El framework está diseñado para detectar y tratar fallos a nivel de aplicación. Esto permite ofrecer servicios con alta disponibilidad sobre un cluster de ordenadores (aunque estos tengan fallos).



B.D.A. Workflow example...



Big Opportunities...

Big Opportunities

A study by International Data Corp. cited data from these sources as being particularly ripe for analysis:

SURVEILLANCE FOOTAGE



There is greater opportunity to embed more intelligence

into digital surveillance cameras so that footage can be captured, analyzed and tagged in real time. This can expedite crime investigations, enhance retailers' analysis of consumer traffic patterns and improve military intelligence.

MEDICAL DEVICES



Various kinds of sensors, including some that may be implanted in

the body, have the potential to capture biometrics, track the effectiveness of medicines, correlate bodily activity with health, monitor potential outbreaks of viruses and more, all in real time.

SOCIAL MEDIA



Information freely published by massive groups of people

through online social outlets can be a great source of data to help bring the next big thing to market, predict the outcome of elections, and much more.

CONSUMER IMAGES

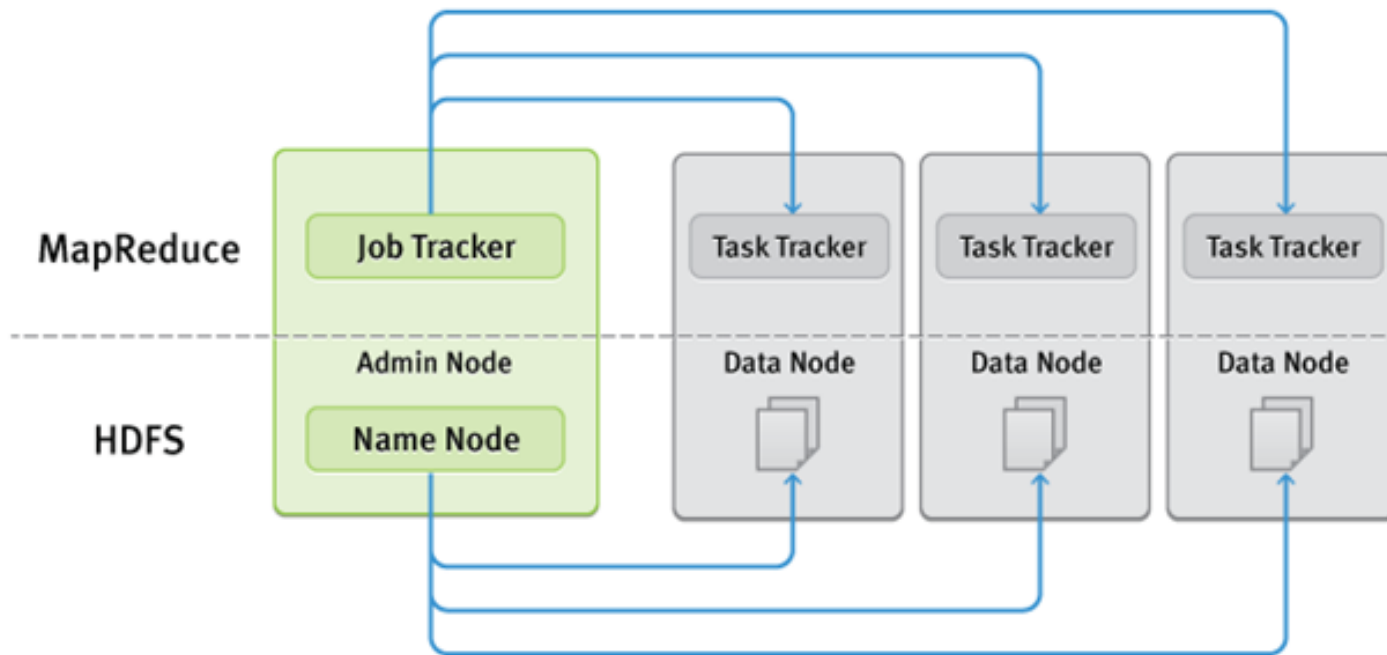
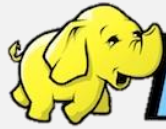


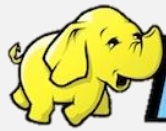
People say a lot about themselves when they post pictures of themselves

or family and friends online. The key will be the introduction of tagging algorithms that can analyze images in real time when pictures are uploaded, or en masse after they are aggregated from various sites.

Source: International Data Corp.

The Wall Street Journal

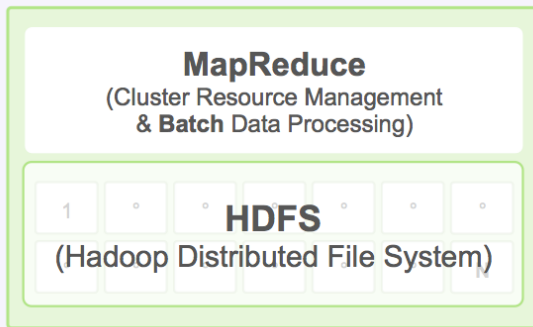




Arquitectura

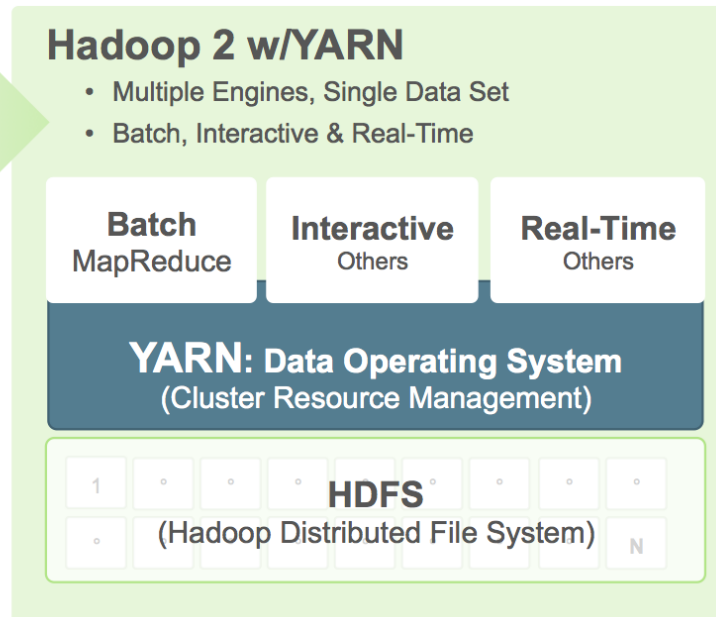
Hadoop 1

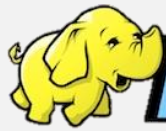
- Silos & Largely batch
- Single Processing engine



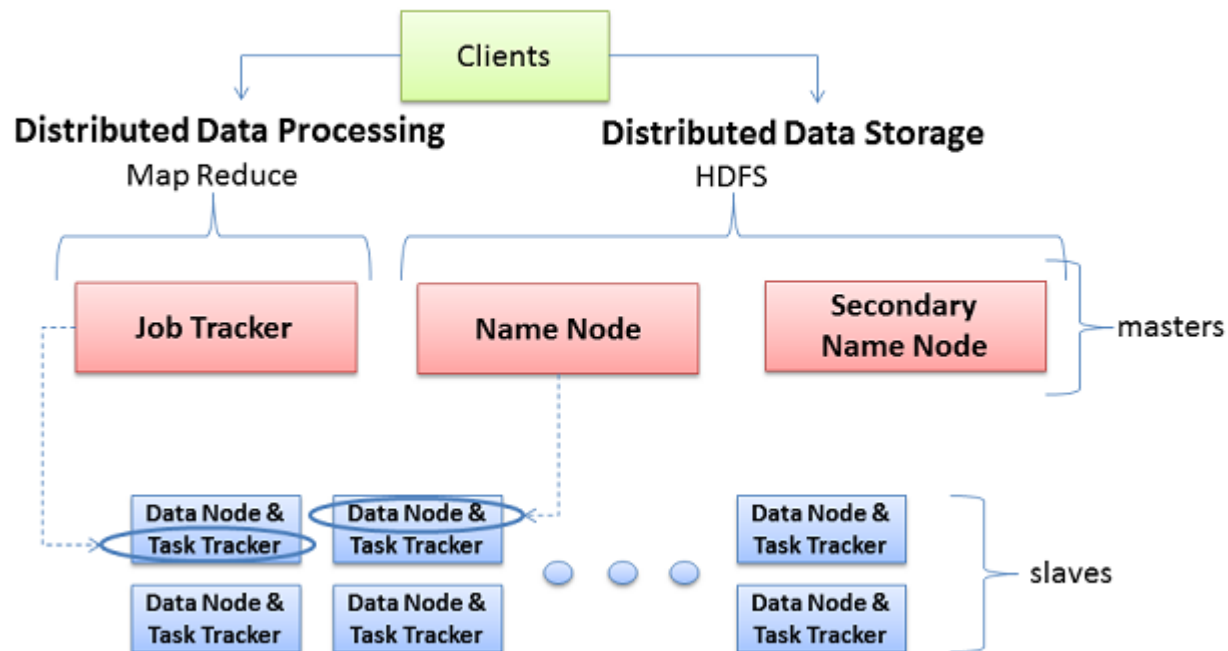
Hadoop 2 w/YARN

- Multiple Engines, Single Data Set
- Batch, Interactive & Real-Time

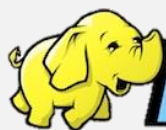




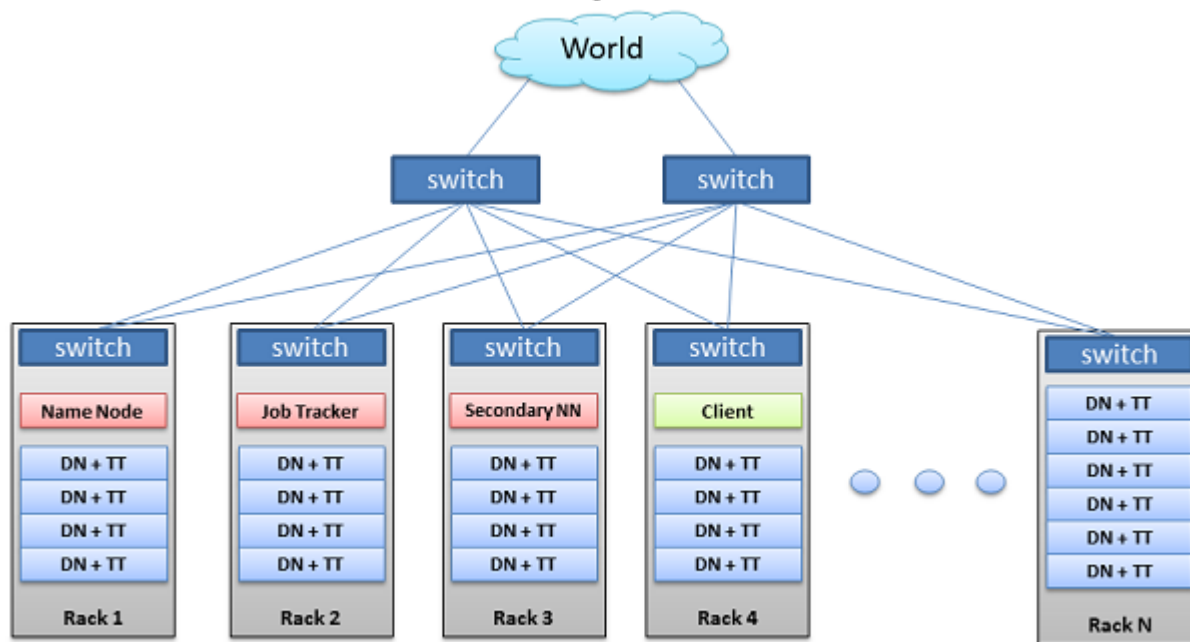
Hadoop Server Roles



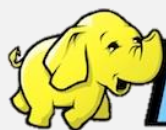
BRAD HEDLUND .com



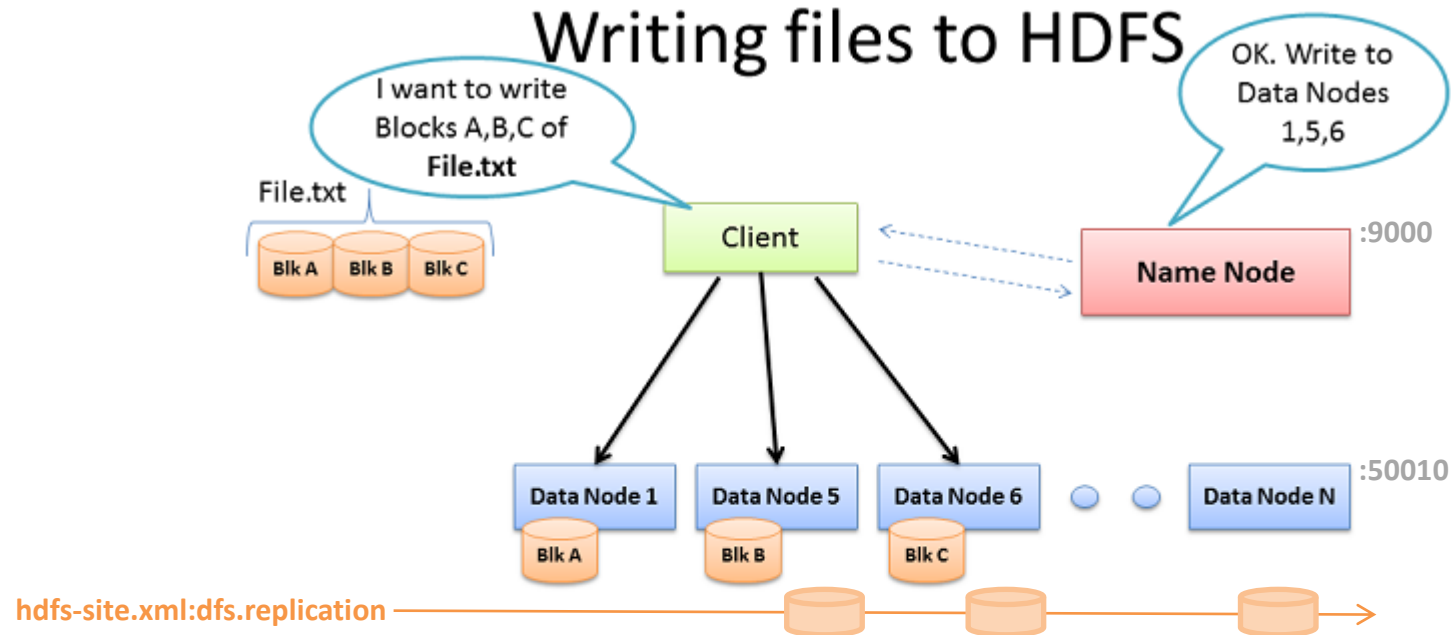
Hadoop Cluster



BRAD HEDLUND .com

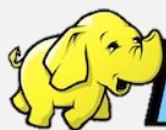


Writing files to HDFS

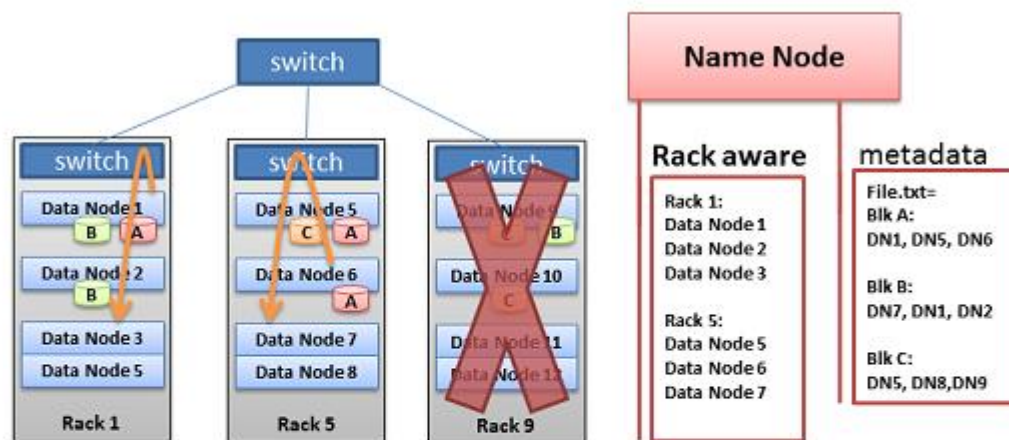


- Client consults Name Node
- Client writes block directly to one Data Node
- Data Nodes replicates block
- Cycle repeats for next block

BRAD HEDLUND .com

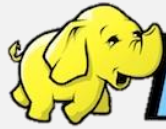


Hadoop Rack Awareness – Why?



- Never loose all data if entire rack fails
- Keep bulky flows in-rack when possible
- Assumption that in-rack is higher bandwidth, lower latency

BRAD HEDLUND .com



Despliegue

Typical Workflow

- Load data into the cluster (HDFS writes)
- Analyze the data (Map Reduce)
- Store results in the cluster (HDFS writes)
- Read the results from the cluster (HDFS reads)

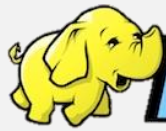
Sample Scenario:

How many times did our customers type the word **"Refund"** into emails sent to customer service?

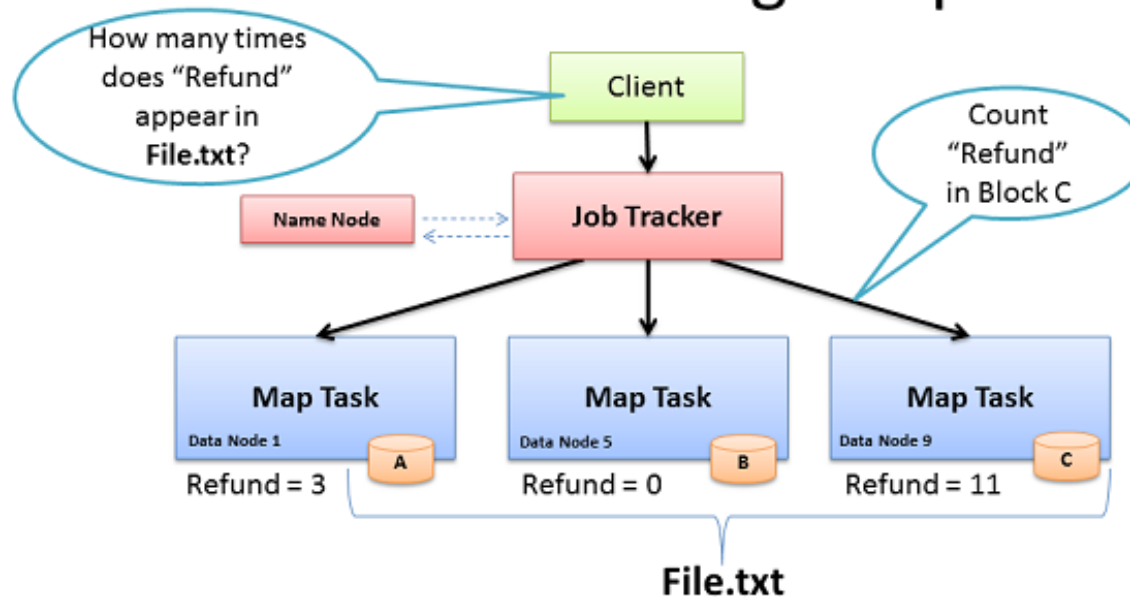
Huge file containing all emails sent
to customer service



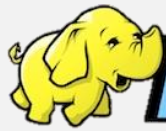
BRAD HEDLUND .com



Data Processing: Map

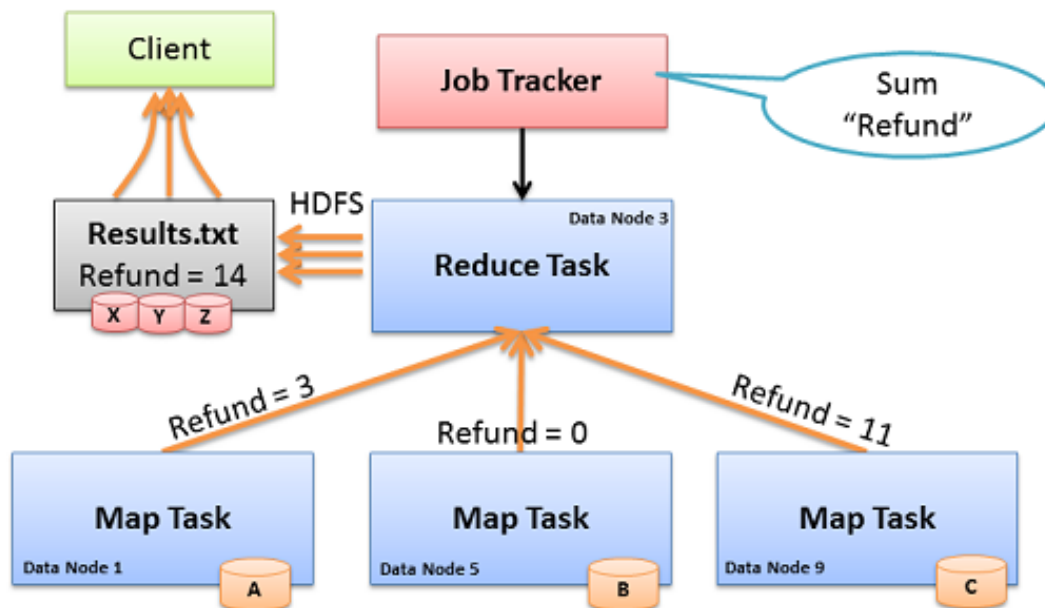


- **Map:** "Run this computation on your local data"
- Job Tracker delivers Java code to Nodes with local data



Despliegue

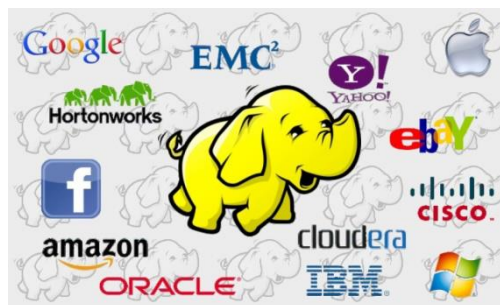
Data Processing: Reduce



- **Reduce:** "Run this computation across Map results"
- Map Tasks send output data to Reducer over the network
- Reduce Task data output written to and read from HDFS

Contenidos

Big
Big
Big
Big
Big

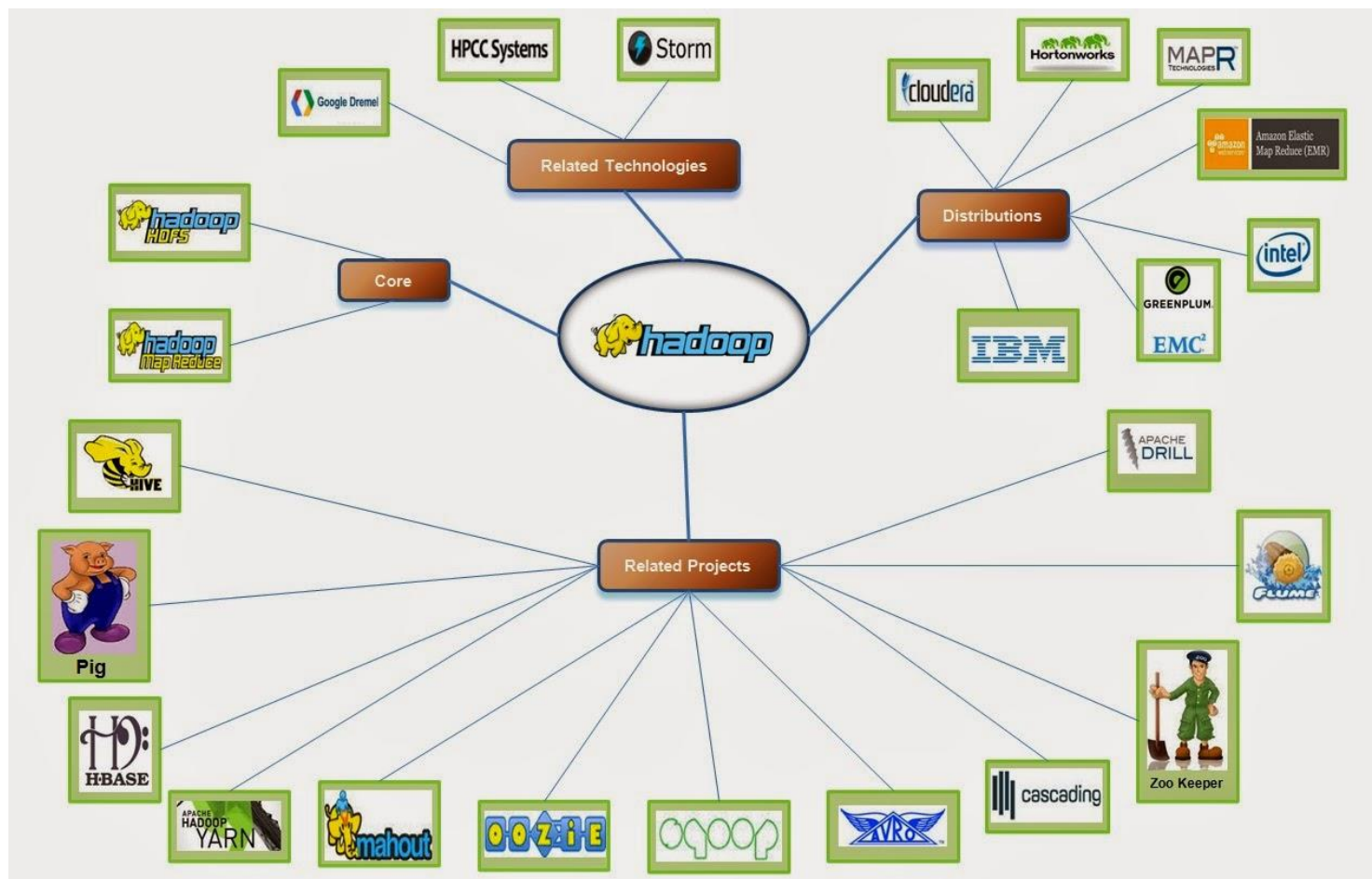


- Inicio
- Transición
- (re)Evolución
- (re)Retos

- Inicio
- Hadoop
- **Ecosistema**
- Evolución

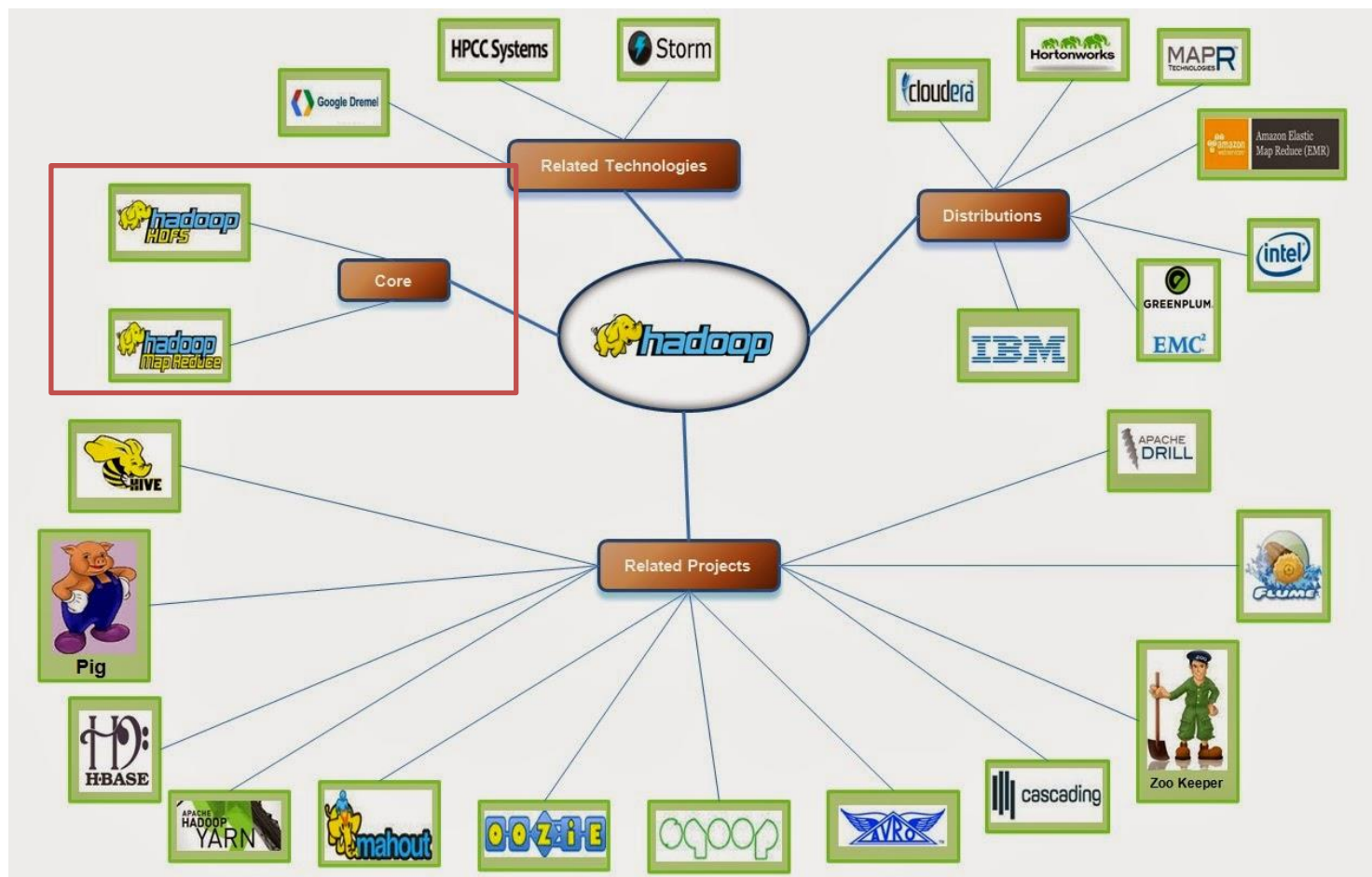


Ecosistema

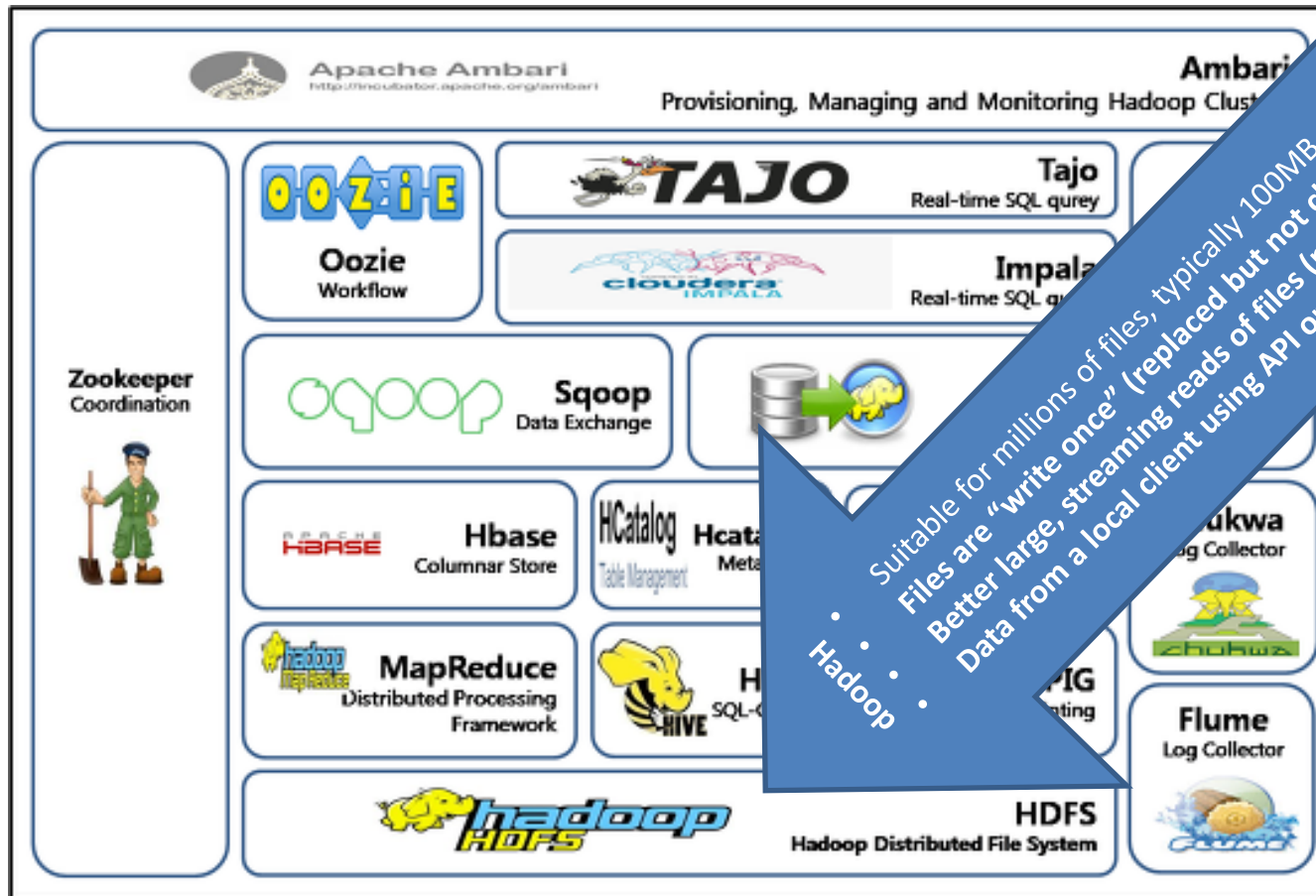




Ecosistema



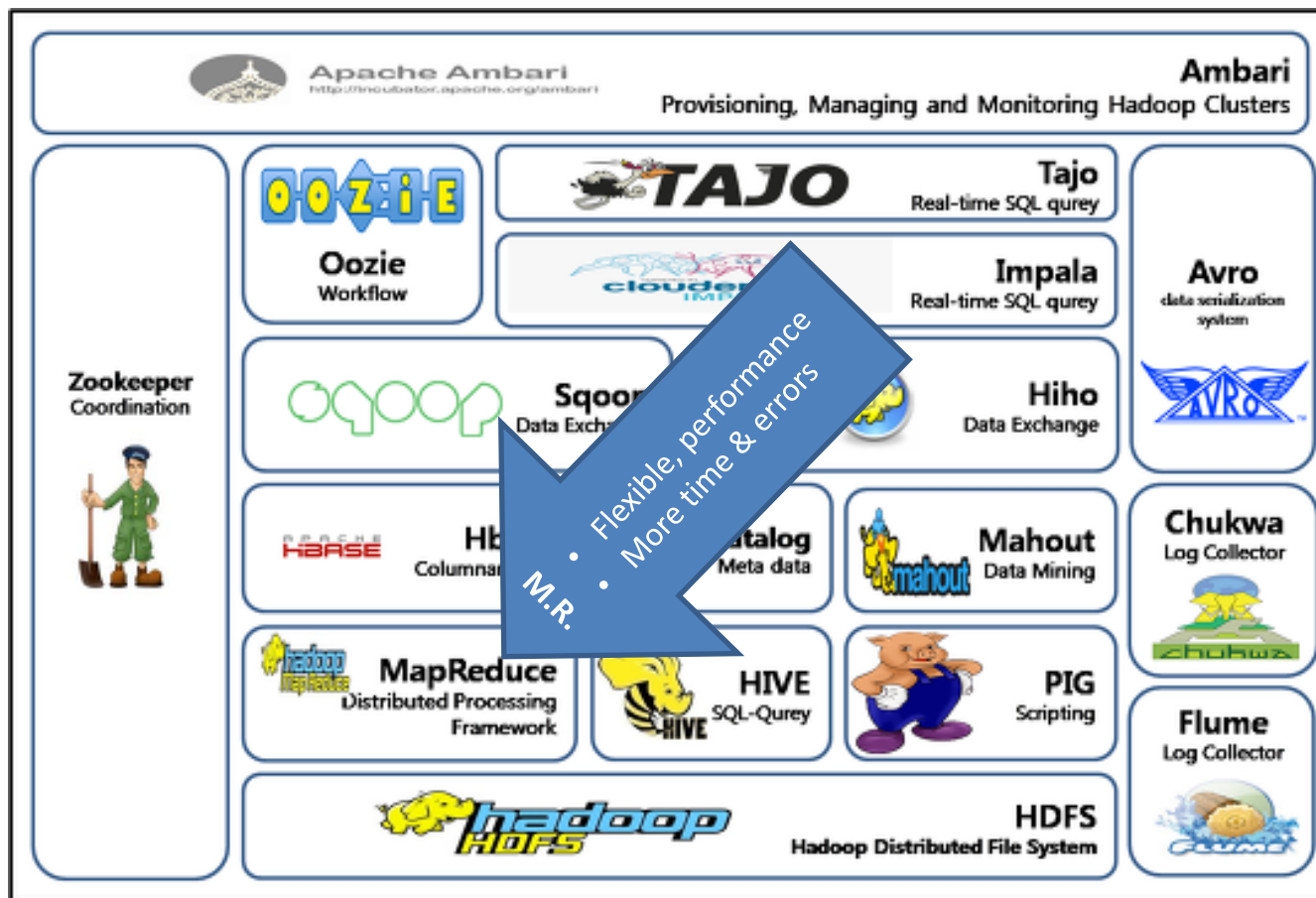
Almacenamiento



Suitable for millions of files, typically 100MB or more
Files are "write once" (replaced but not changed)
Better large, streaming reads of files (not random reads)
Data from a local client using API or command line

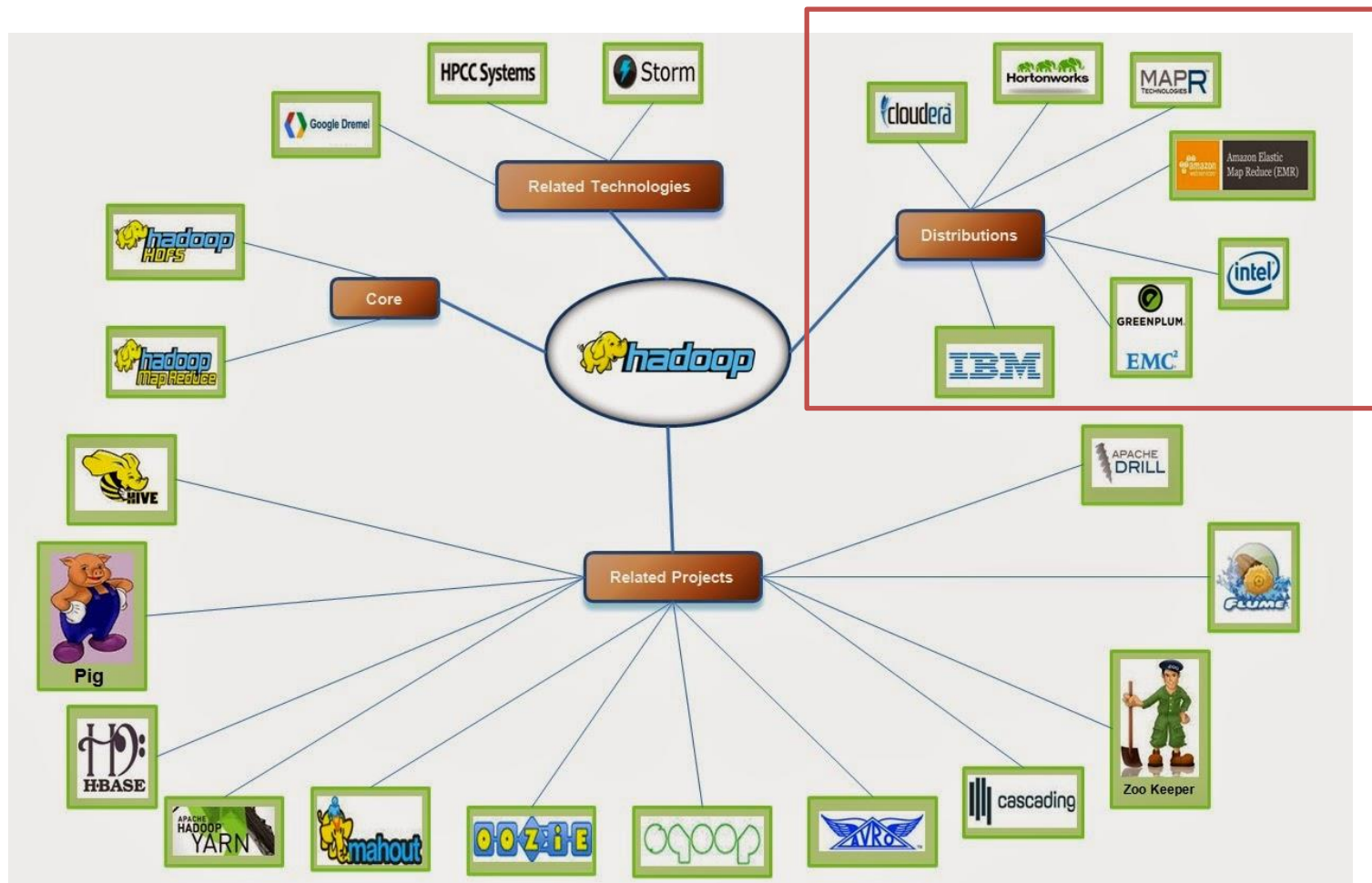
- Hadoop
- Pig
- Hive

Procesamiento



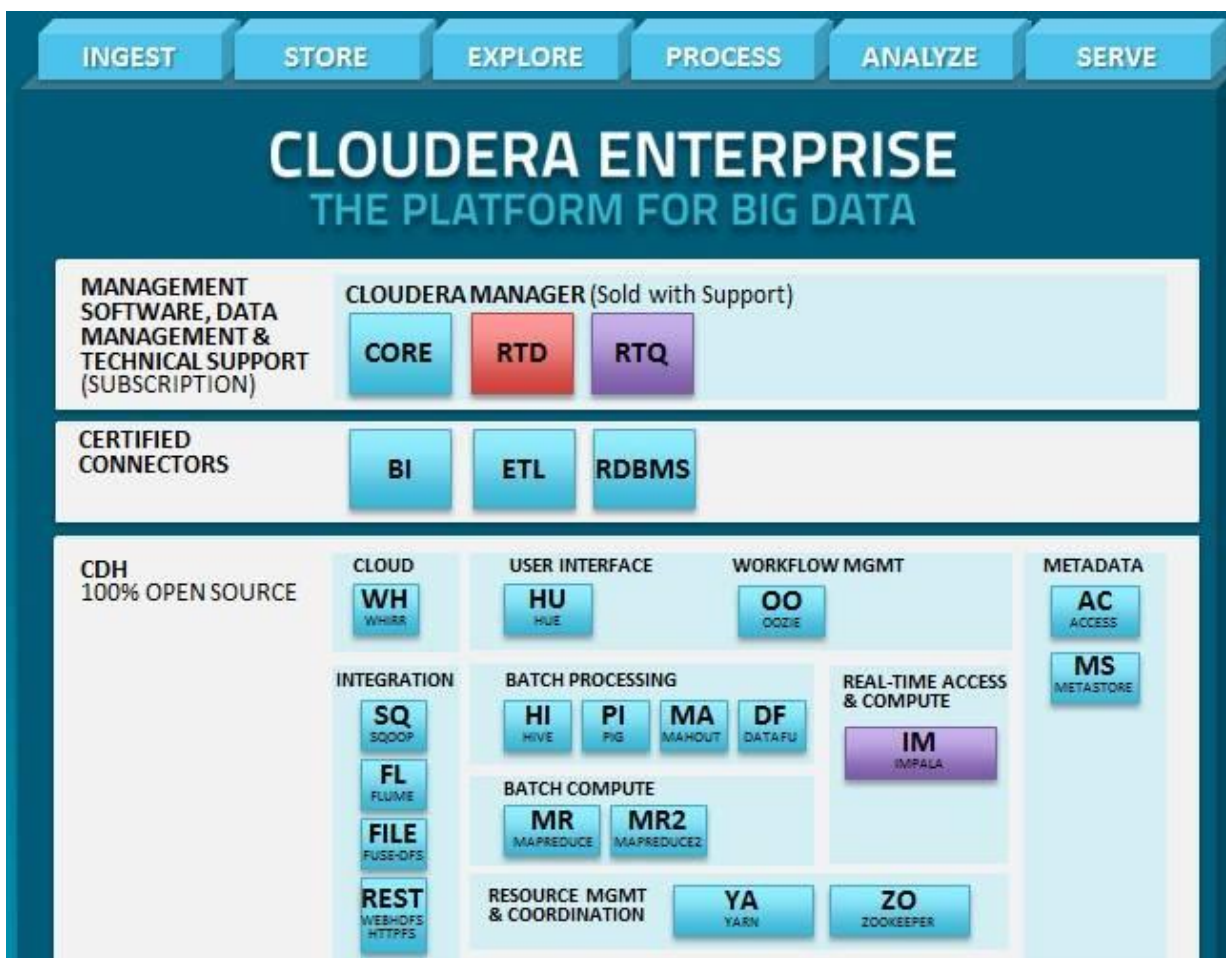


Ecosistema



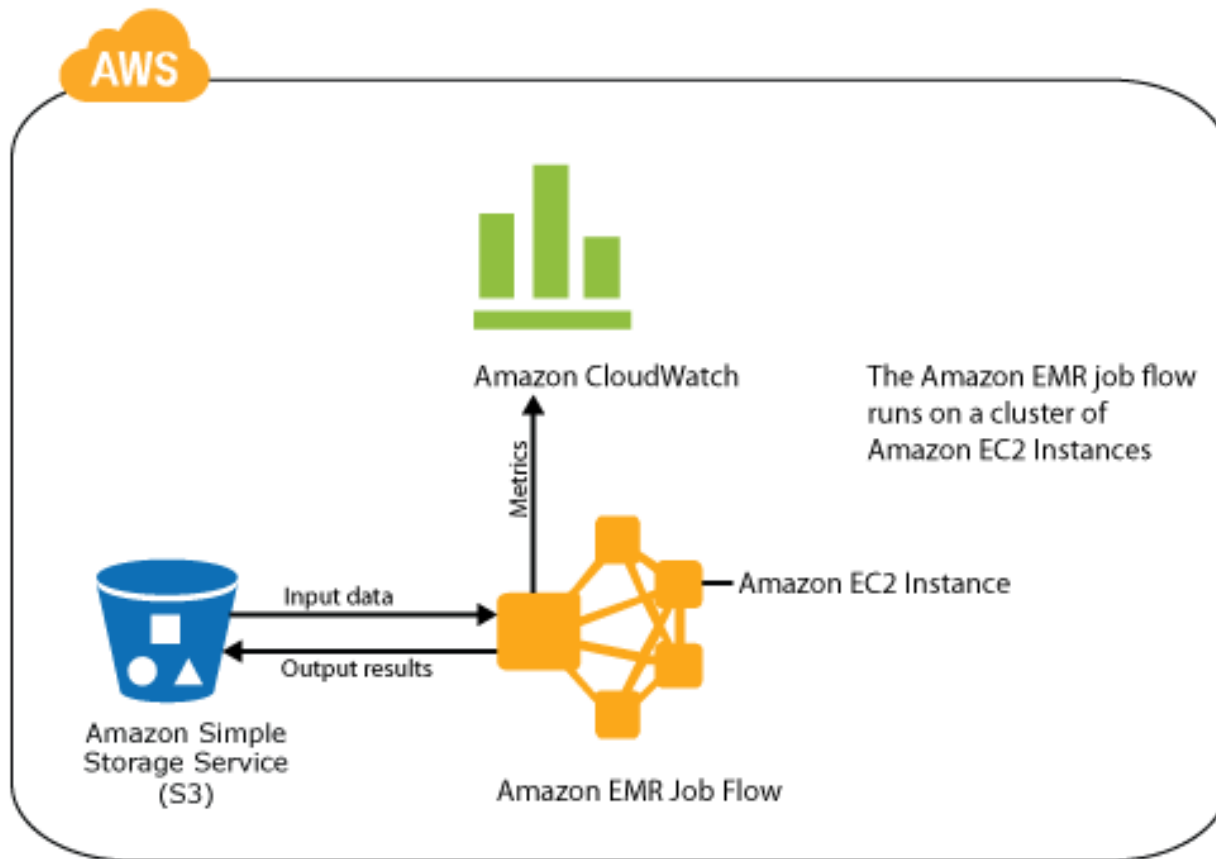


CDH (Cloudera Hadoop Distribution)



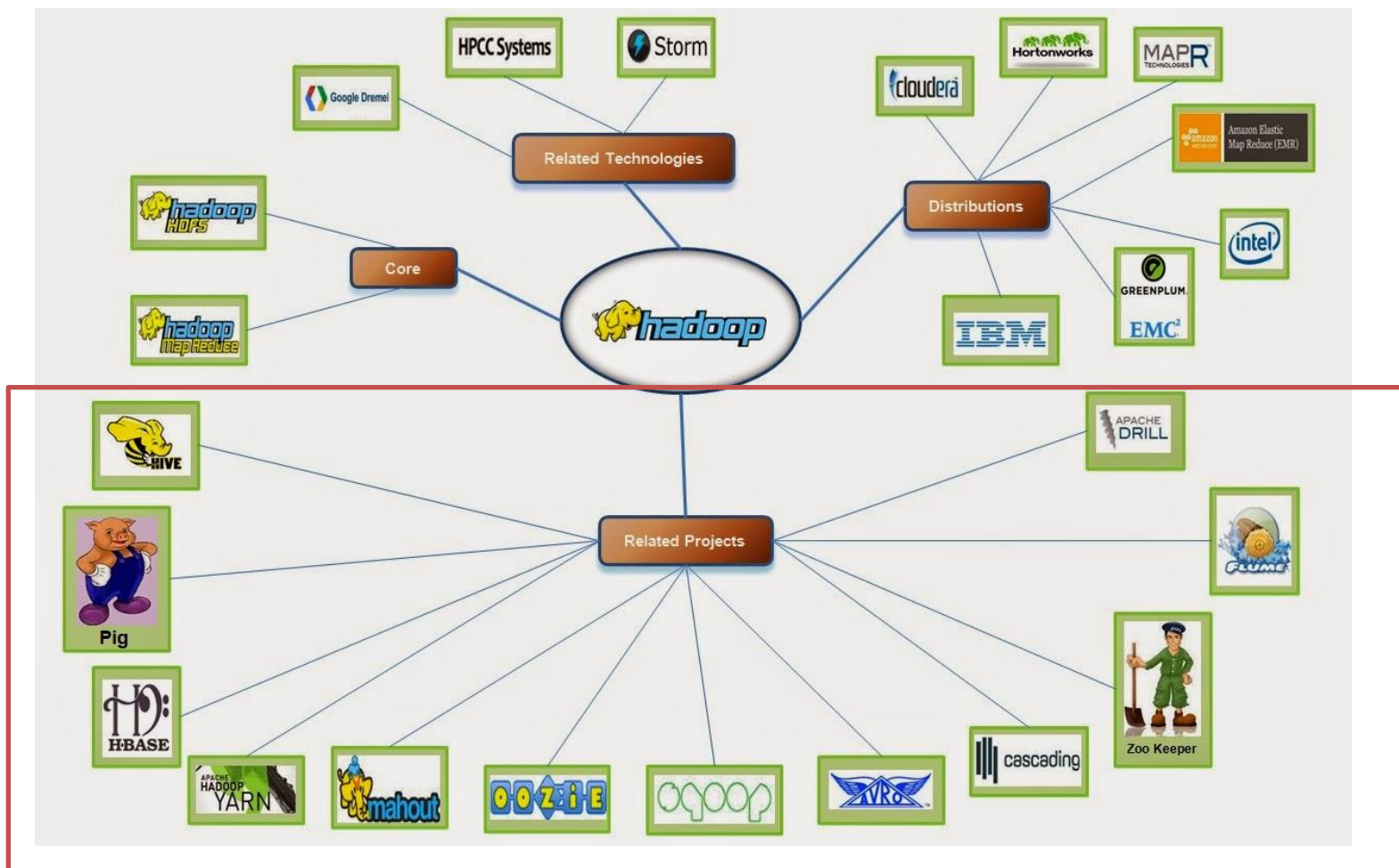


AEMR (Amazon Elastic MapReduce)





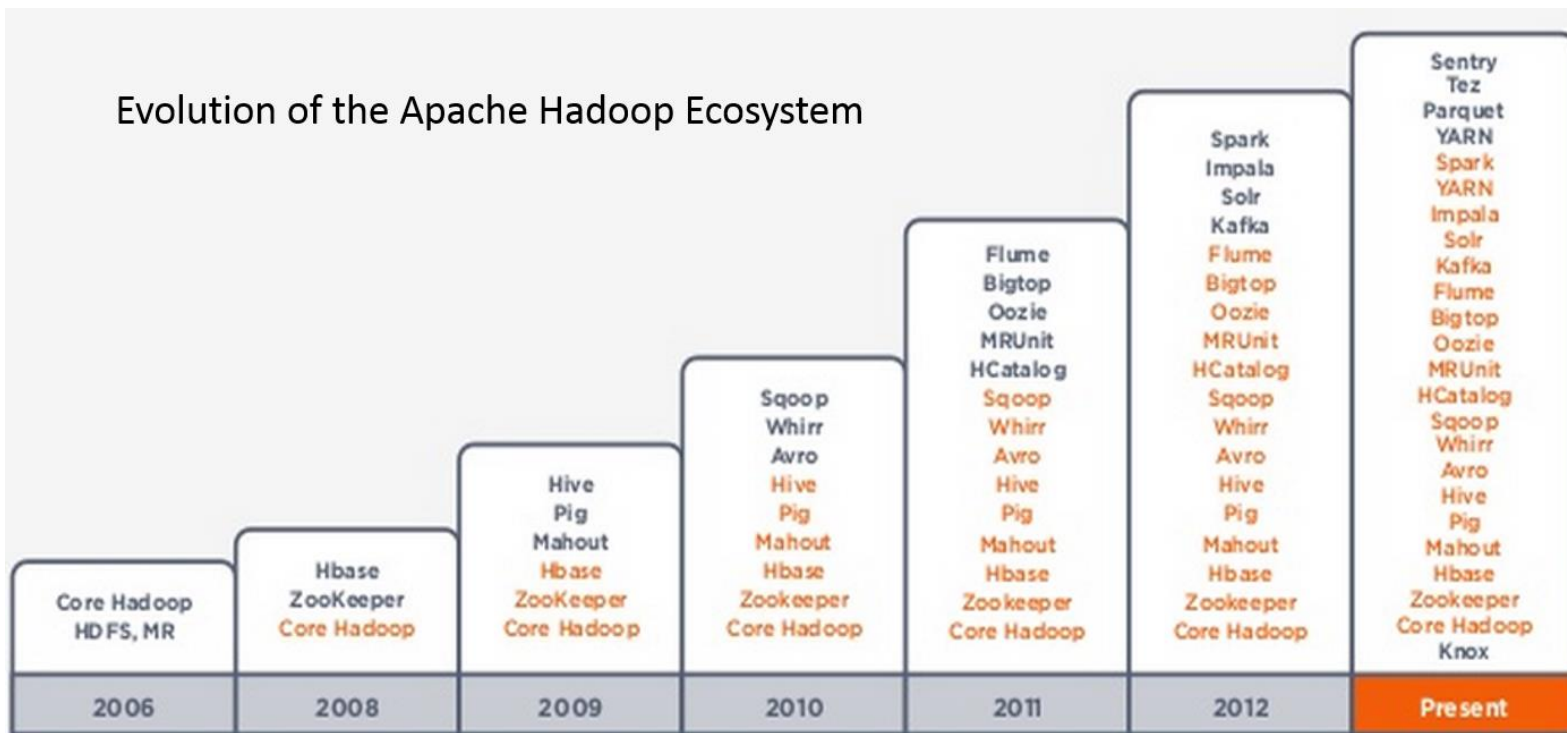
Ecosistema





Ecosistema

Evolution of the Apache Hadoop Ecosystem

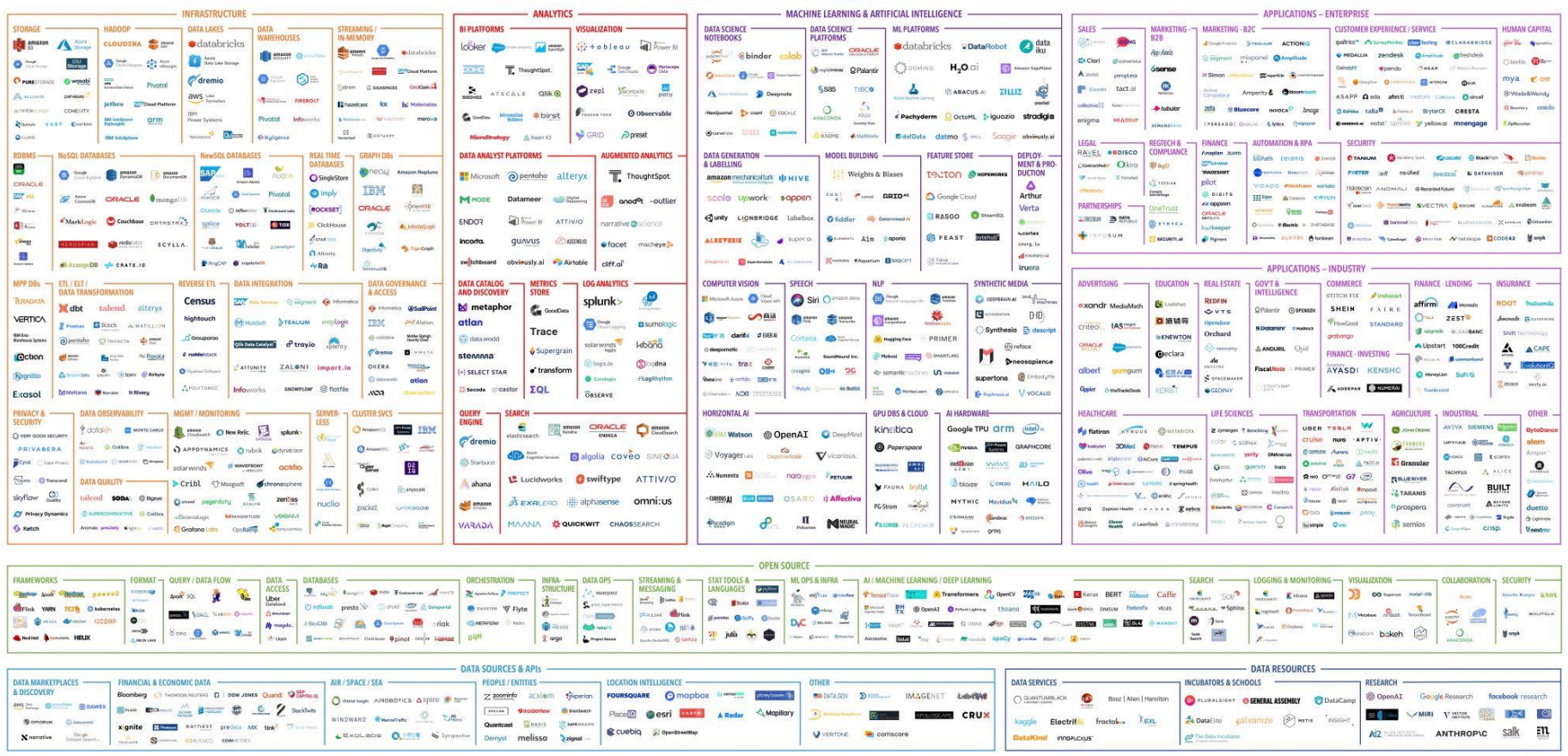




Ecosistema

(funcionalidad)

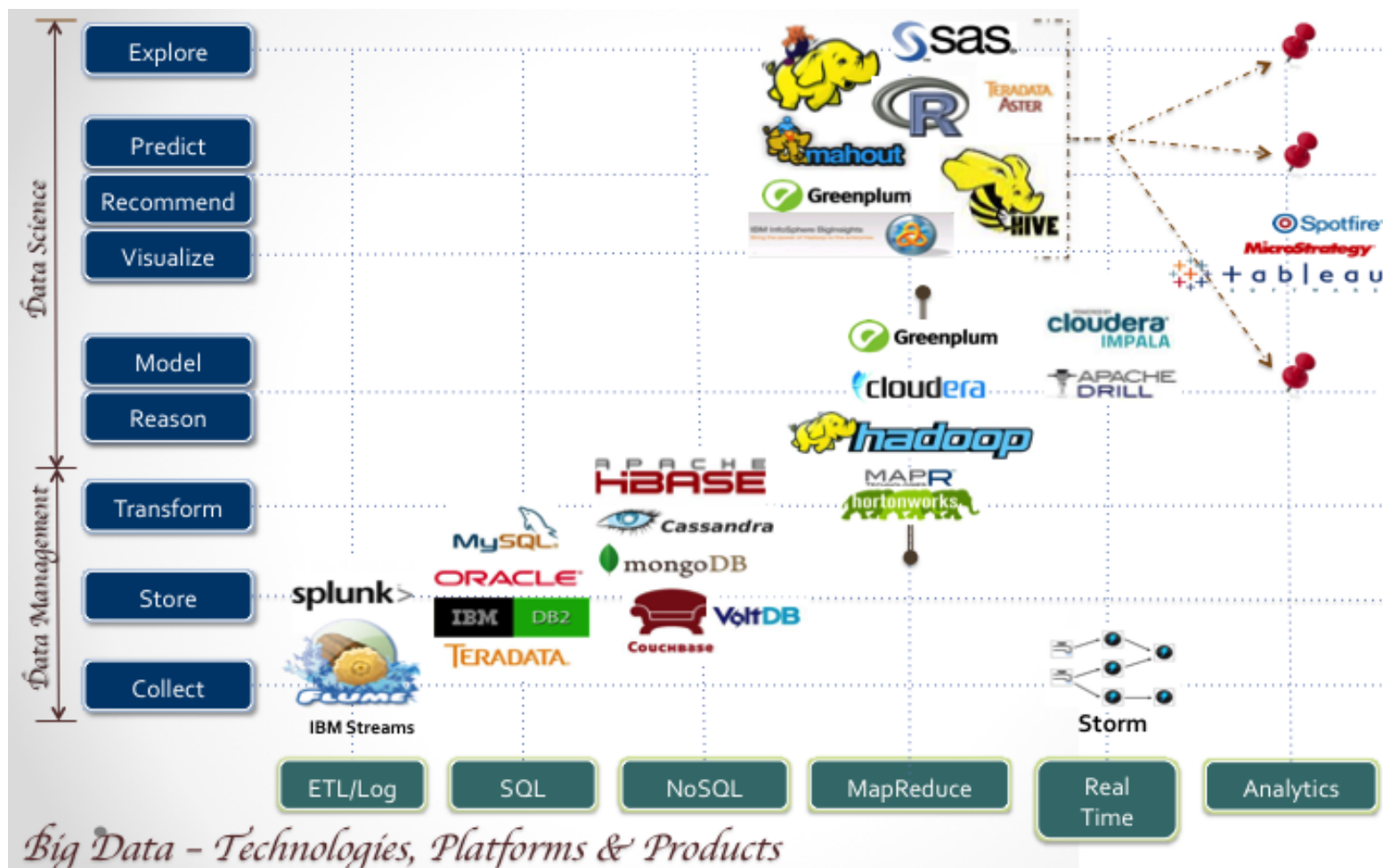
MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, AND DATA (MAD) LANDSCAPE 2021





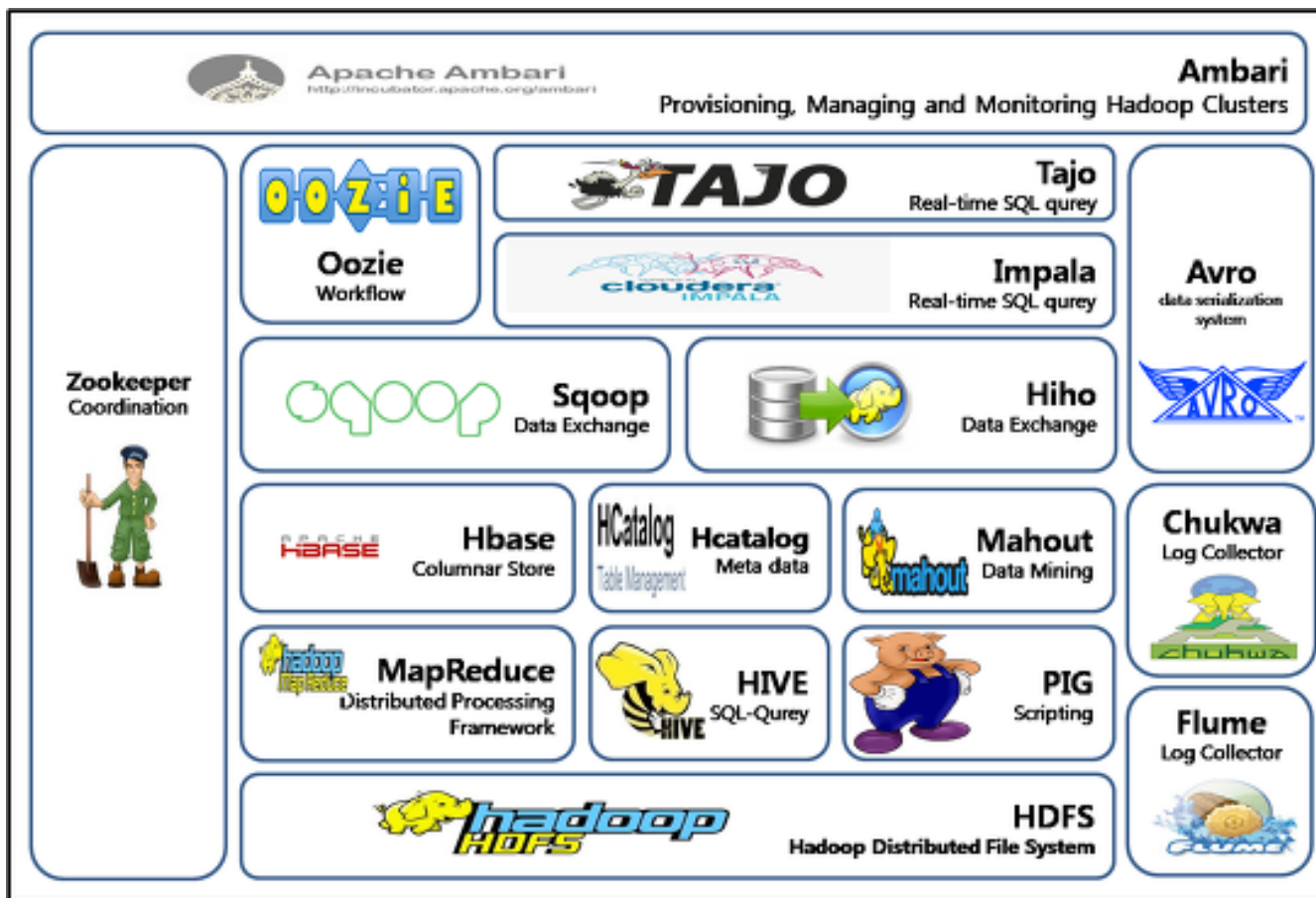
Ecosistema

(funcionalidad)

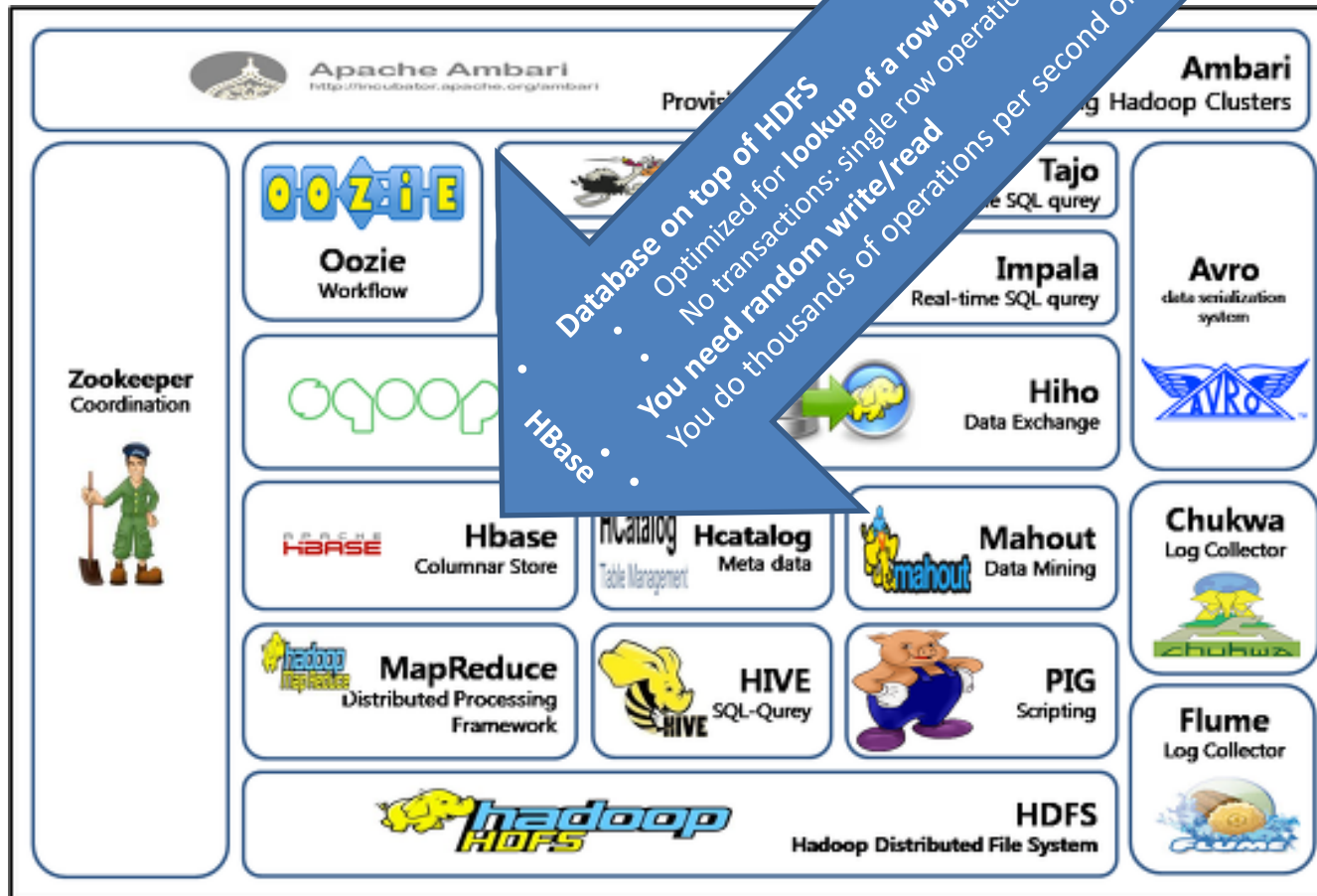




Ecosistema (estructura)



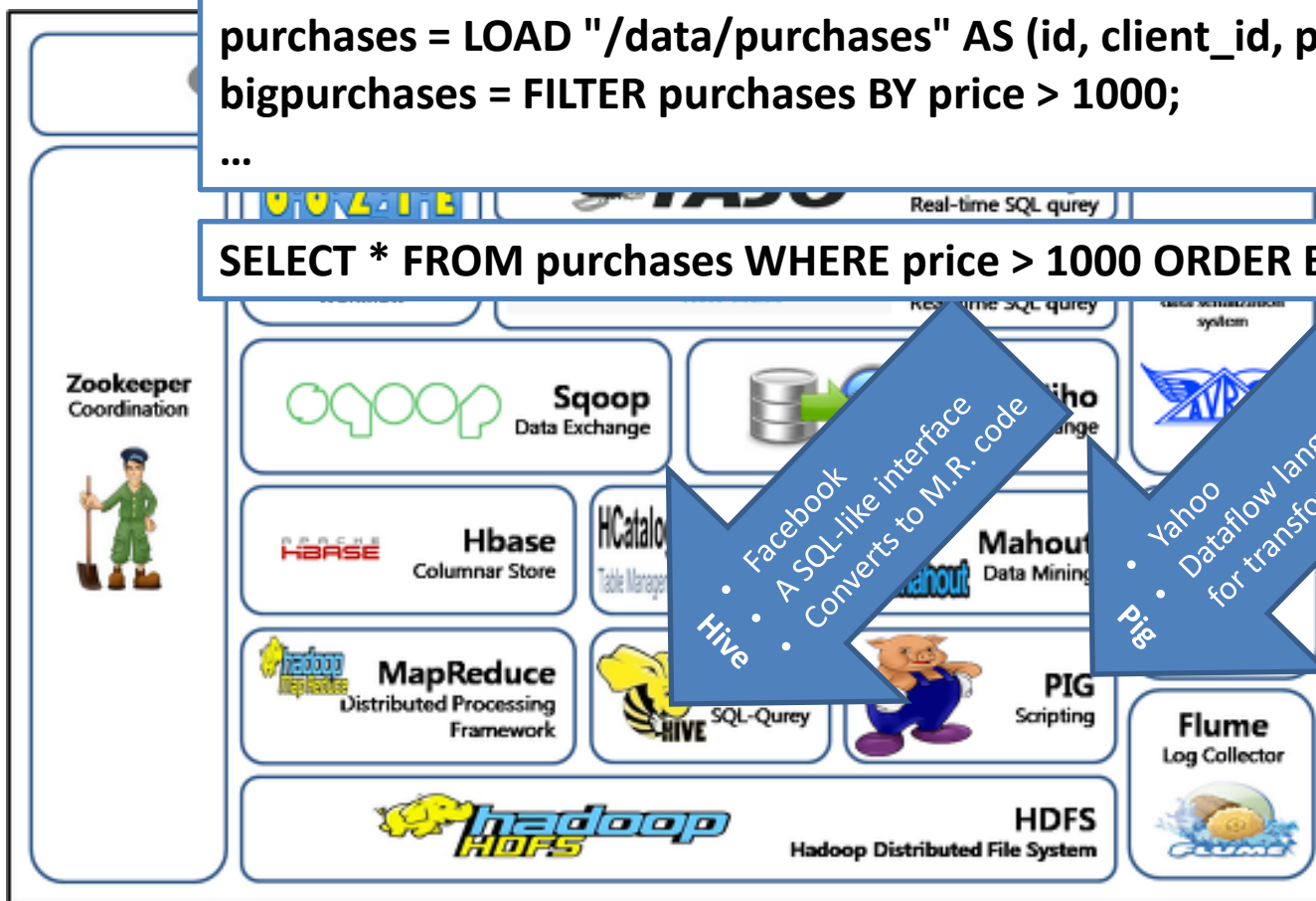
Almacenamiento



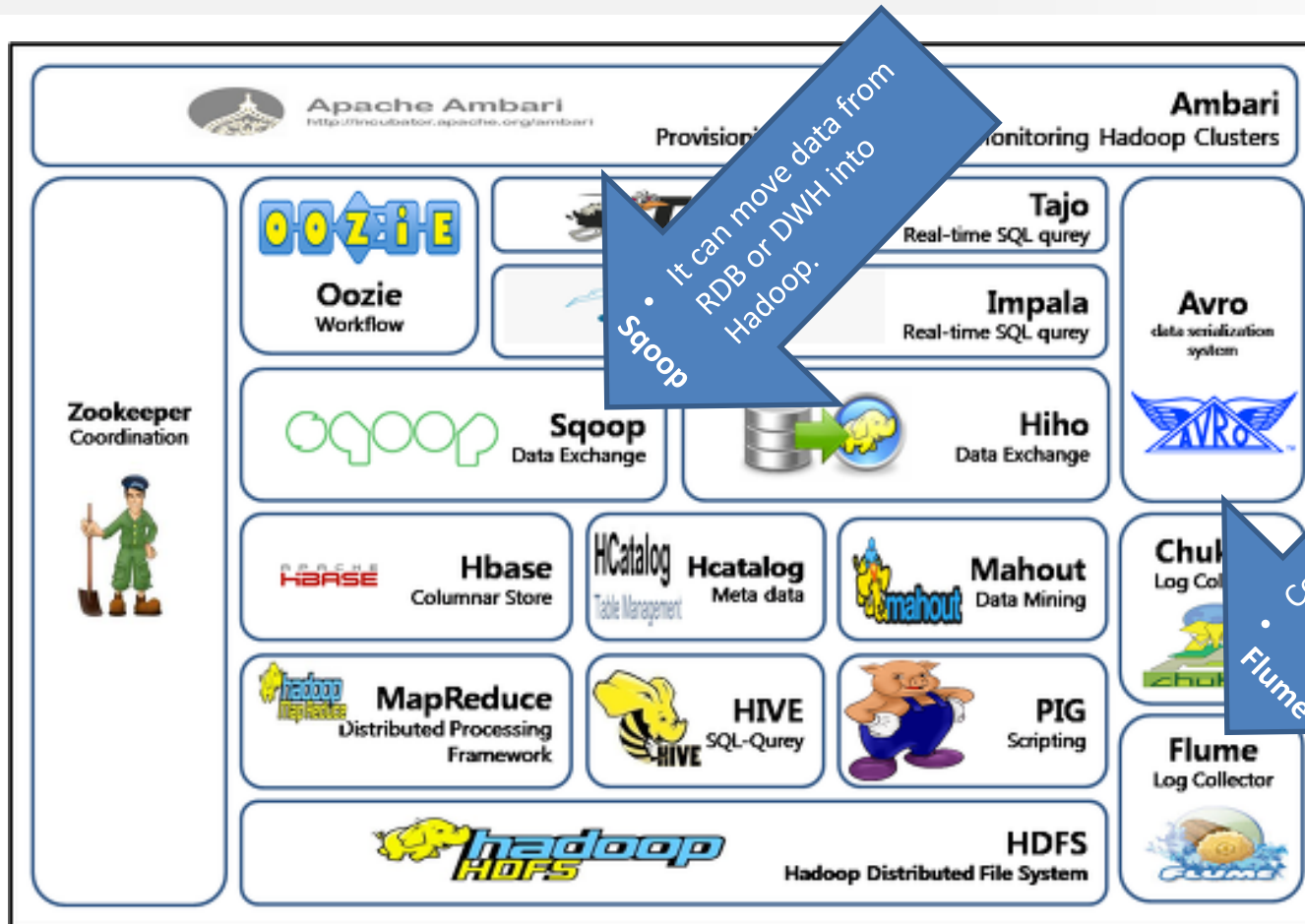
Procesamiento

```
purchases = LOAD "/data/purchases" AS (id, client_id, product_id);  
bigpurchases = FILTER purchases BY price > 1000;  
...
```

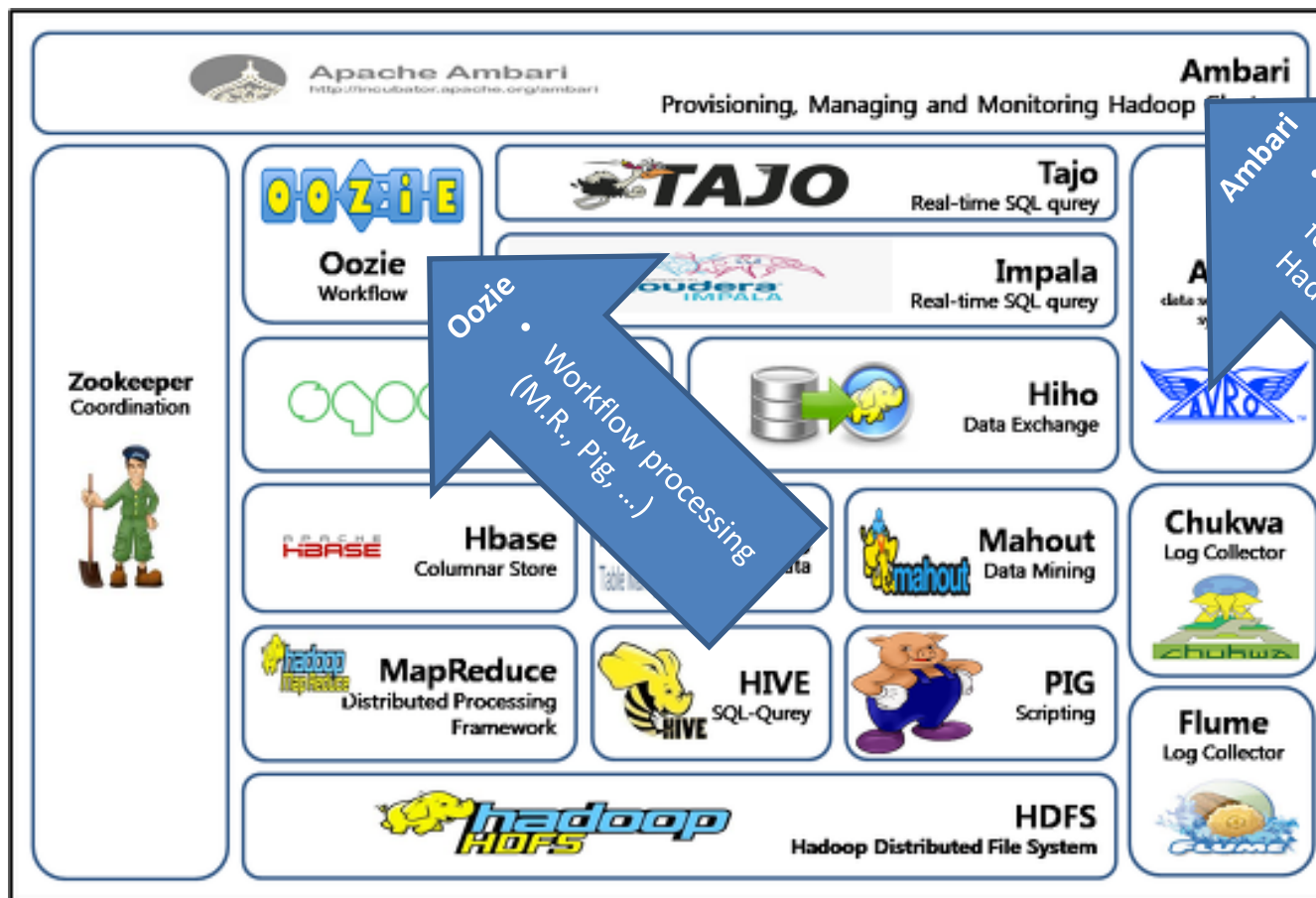
```
SELECT * FROM purchases WHERE price > 1000 ORDER BY client_id
```



Almacenamiento: Integración



Procesamiento: orquestación

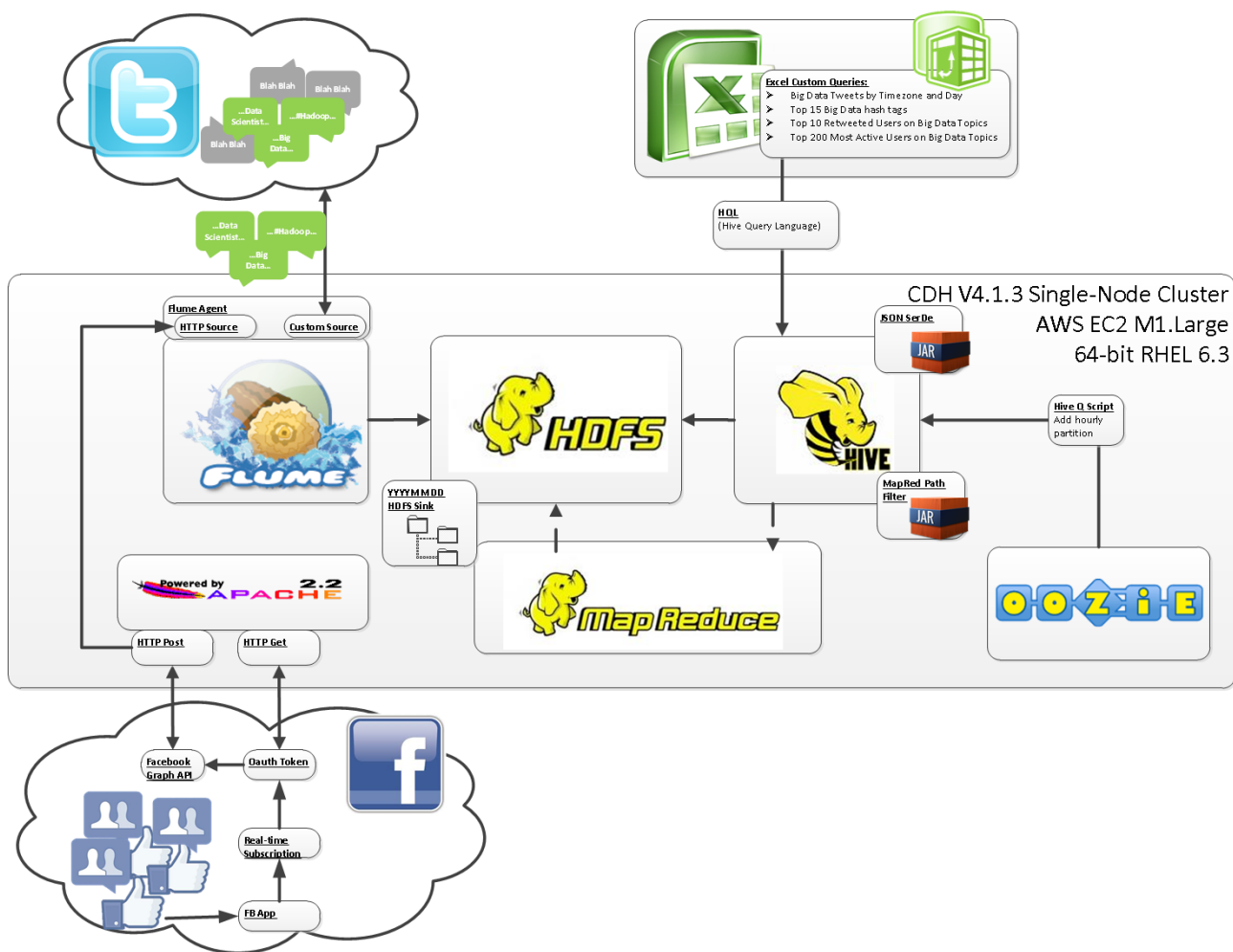


Ambari

- Hortonworks
- A web-based tool-set for managing Apache Hadoop clusters

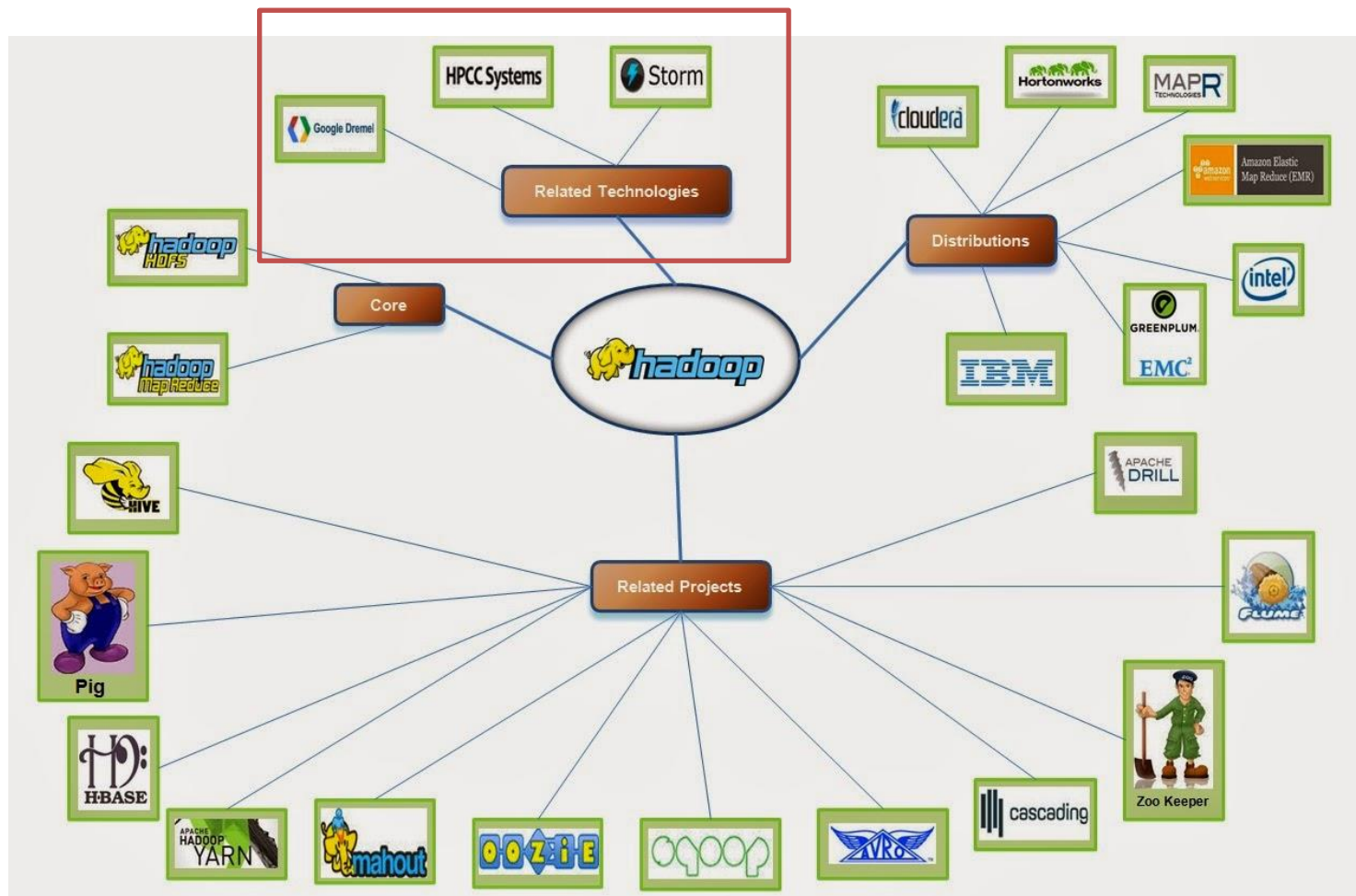


Ej.: Flume-Hive-oozie





Ecosistema



Contenidos

Big
Big
Big
Big
Big

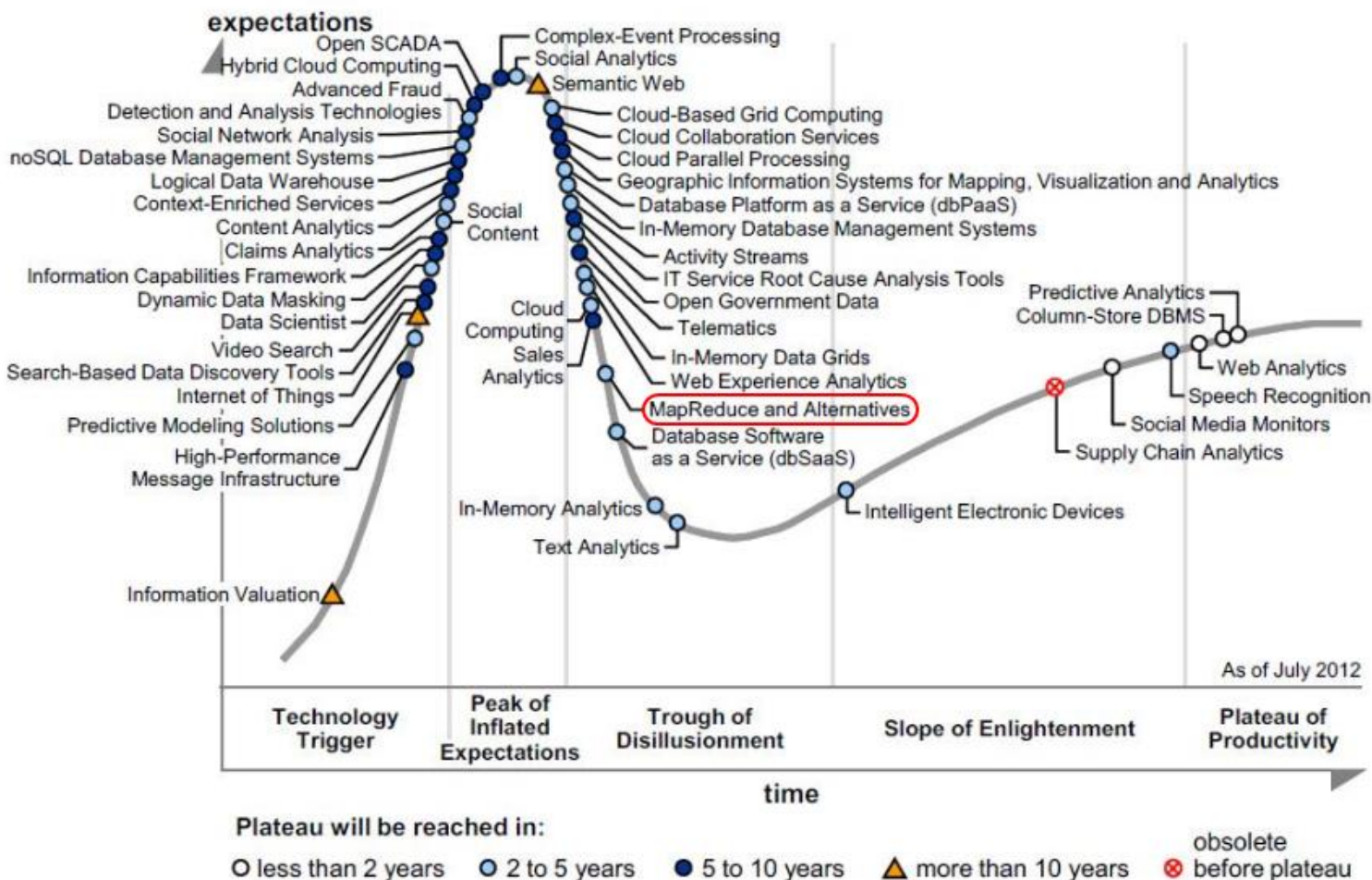


- Inicio
- Transición
- (re)Evolución
- (re)Retos
- Inicio
- Hadoop
- Ecosistema
- **Evolución**

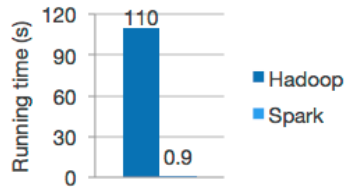


Alternativas

Figure 1. Hype Cycle for Big Data, 2012



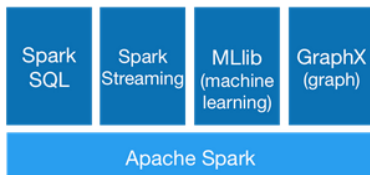
Source: Gartner (July 2012)



Logistic regression in Hadoop and Spark

```
file = spark.textFile("hdfs://...")  
  
file.flatMap(lambda line: line.split())  
    .map(lambda word: (word, 1))  
    .reduceByKey(lambda a, b: a+b)
```

Word count in Spark's Python API



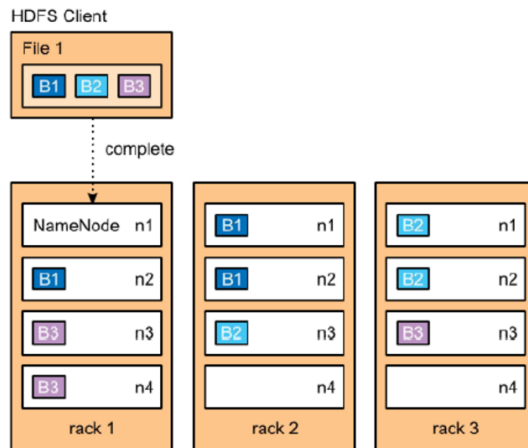
- Velocidad
 - Se evita en lo posible escribir en disco los resultados intermedios
- Facilidad de uso
 - Transformaciones y operaciones
- Polivalente
 - En Scala, Python, Java
 - Combina SQL, *streaming*, analítico...
 - Trabaja con HDFS, HBase, S3, ...

Spark vs. Hadoop

	Hadoop	Spark
Datos	Resultados intermedios en disco	Mantener en memoria todo lo posible
Tolerancia a fallos en datos	HDF (2+1)	RDD
Procesamiento	Trabajos map/reduce	DAG
Programación	Java + Otros con tuberías	Scala, Python, Java, etc.
Escenario	Trabajos lentos en <i>batch</i>	<i>Batch</i> + Real-time + Iterativo + Interactivo
	(Más) E/S de disco + red que Spark	(Más) memoria RAM que Hadoop
	Marco de trabajo de propósito general	Alternativa más que replazamiento de Hadoop

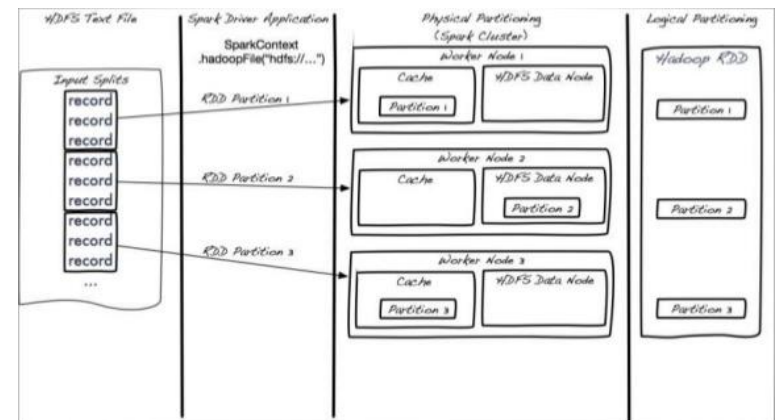
Spark vs. Hadoop

HDFS (HDFS + 3copies)



- Datos leídos en bloques
 - 64MB por defecto
- Usados como en ficheros

RDD (Resilient Distributed Dataset)



- Datos leídos como RDD
 - Inmutable, regenerable, en memoria
- DataFrame = RDD[filas] + Schema

Spark

vs.

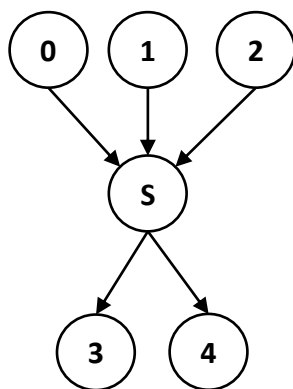


M-R
(Map-Reduce)

~MPI_Scatter

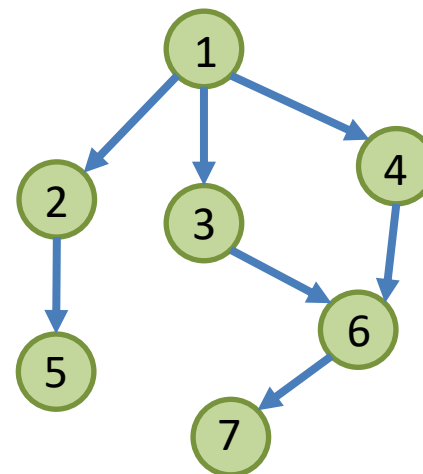
~MPI_all-to-all

~MPI_Gather



- DAG con 2 estados
 - Map + Reduce
- Datos intermedios a disco

DAG
(Directed Acyclic Graph)



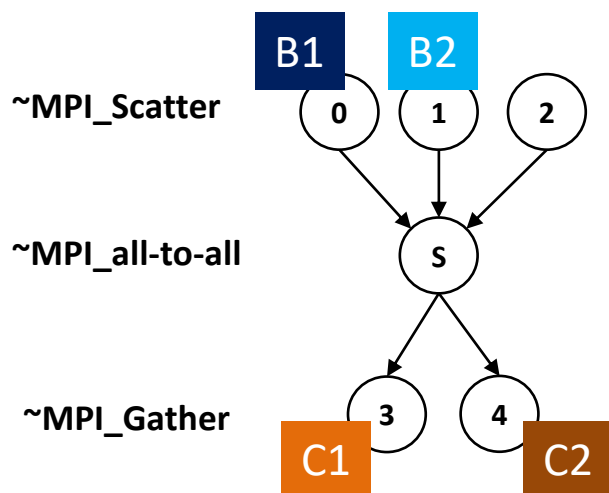
- DAG con N estados
- Datos intermedios en memoria



vs.

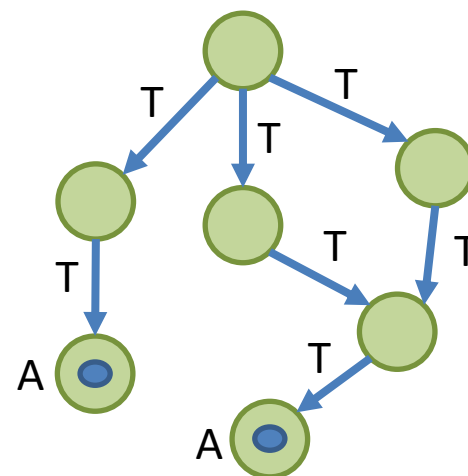


HDFS + M-R

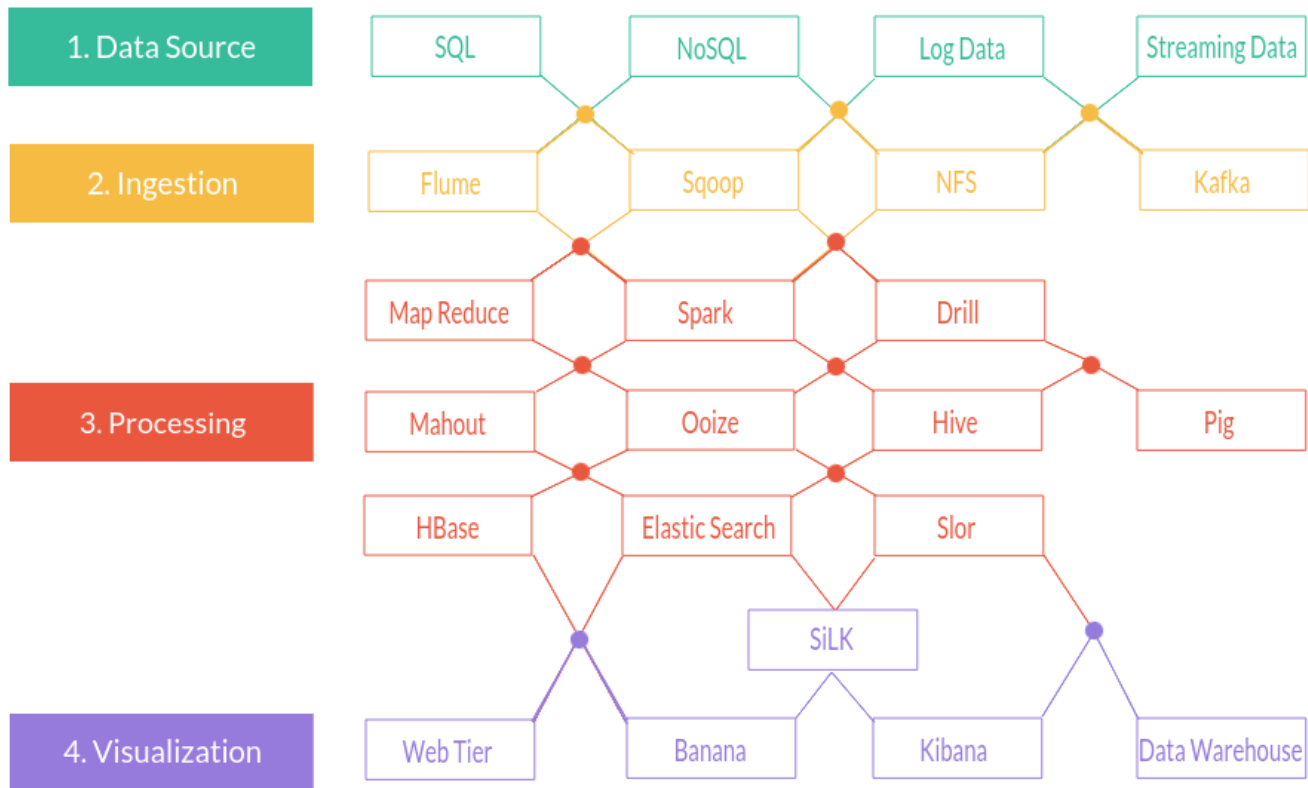


- Operaciones sobre bloques:
 - En M se leerán
 - En R se escribirán

RDD + DAG



- Dos tipos operaciones:
 - Transformaciones (RDD[] -> RDD'[])
 - Acciones (RDD[] -> valor)





1. `val conf = new SparkConf().setMaster("local[2]")`
2. `val sc = new SparkContext(conf)`
3. `val lines = sc.textFile(path, 2)`
4. `val words = lines.flatMap(_.split(" "))`
5. `val pairs = words.map(word => (word, 1))`
6. `val wordCounts = pairs.reduceByKey(_ + _)`
7. `val localValues = wordCounts.take(100)`
8. `localValues.foreach(r => println(r))`



```
1. val conf = new SparkConf().setMaster("local[2]")
2. val sc = new SparkContext(conf)
3. val lines = sc.textFile(path, 2)
4. val words = lines.flatMap(_.split(" "))
5. val pairs = words.map(word => (word, 1))
6. val wordCounts = pairs.reduceByKey(_ + _)
7. val localValues = wordCounts.take(100)
8. localValues.foreach(r => println(r))
```



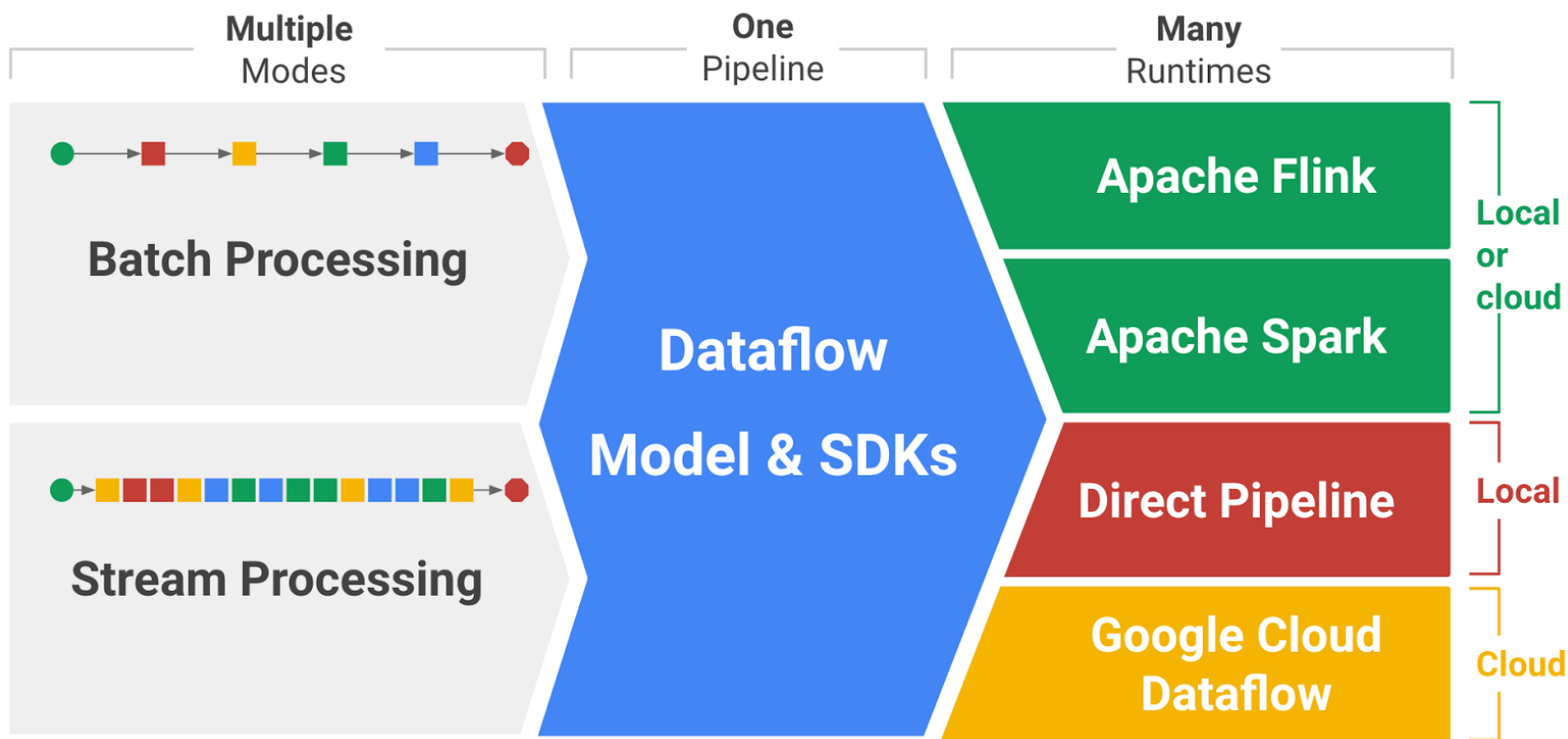
```
1. val conf = new SparkConf().setMaster("local[2]")
2. val sc = new SparkContext(conf)
3. val lines = sc.textFile(path, 2)
4. val words = lines.flatMap(_.split(" "))
5. val pairs = words.map(word => (word, 1))
6. val wordCounts = pairs.reduceByKey(_ + _)
7. val localValues = wordCounts.take(100)
8. localValues.foreach(r => println(r))
```

Google Dataflow



- Data pipelines for helping to ingest, transform, and analyze data.

Google Dataflow



Google Dataflow



```
Pipeline p = Pipeline.create();  
p.begin();
```

```
.apply(TextIO.Read.from("gs://..."))
```

```
.apply(ParDo.of(new ExtractTags()))
```

```
.apply(Count.create())
```

```
.apply(ParDo.of(new ExpandPrefixes()))
```

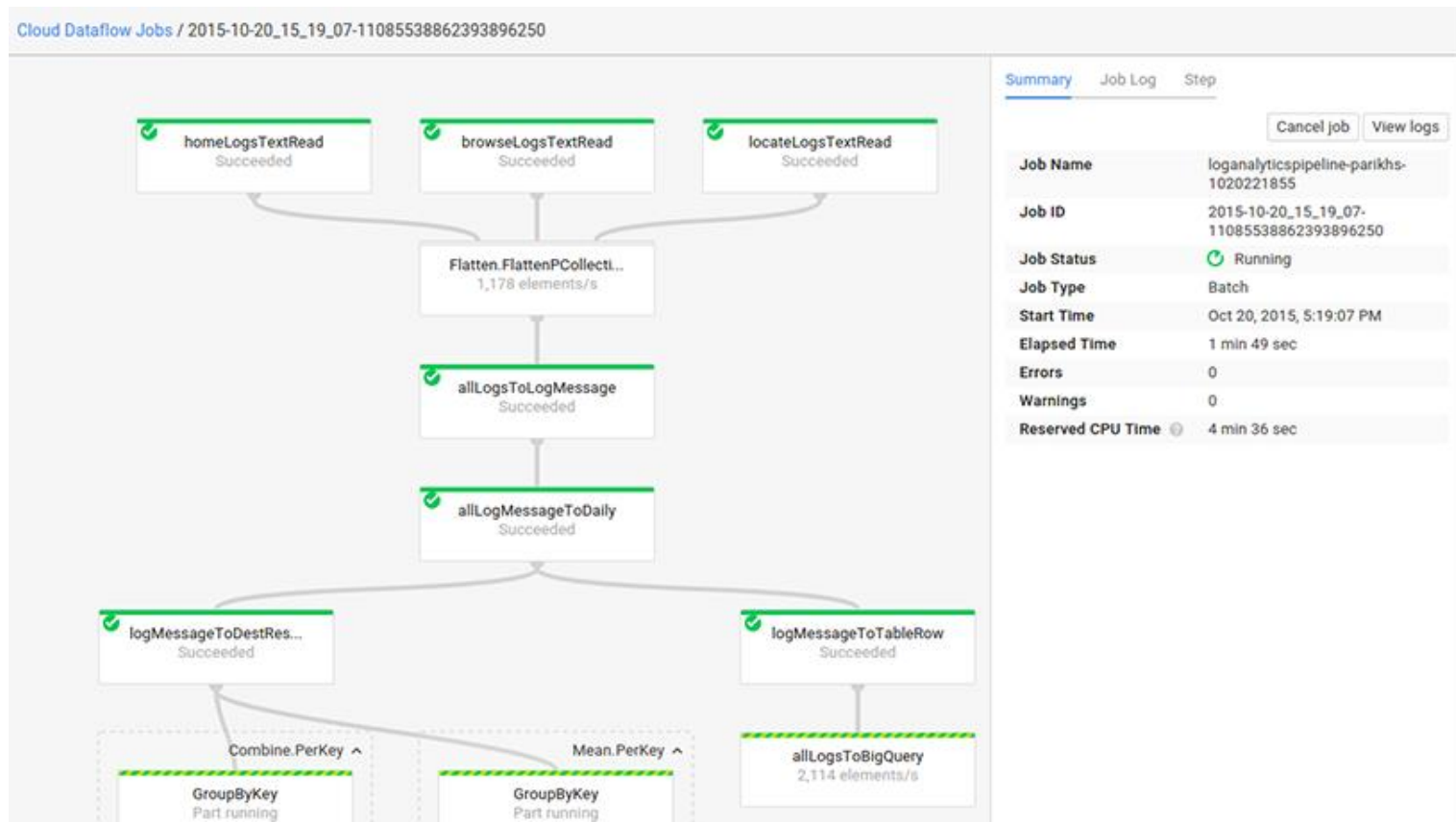
```
.apply(Top.largest
```

```
.apply(TextIO.Wri
```

```
p.run();
```

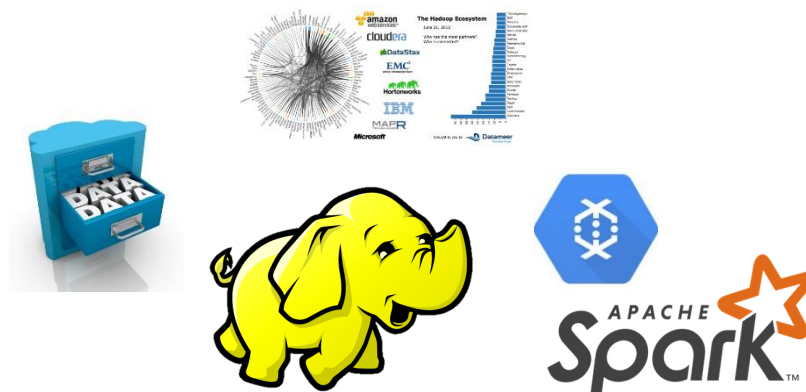
```
class ExpandPrefixes ... {  
    ...  
    public void processElement(ProcessContext c) {  
        String word = c.element().getKey();  
        for (int i = 1; i <= word.length(); i++) {  
            String prefix = word.substring(0, i);  
            c.output(KV.of(prefix, c.element()));  
        }  
    }  
}
```

Google Dataflow



Contenidos

Big
Big
Big
Big
Big



- Inicio
- Transición
- (re)Evolución
- (re)Retos

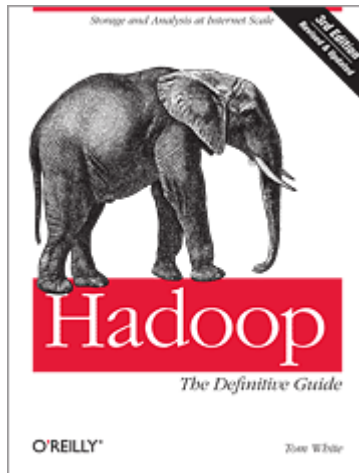
- Inicio
- Hadoop
- Ecosistema
- Evolución

Bibliografía: tutoriales

- Página Web oficial:
 - <http://hadoop.apache.org/>
- Introducción a cómo funciona Hadoop:
 - <http://blog.csdn.net/suifeng3051/article/details/17288047>
- Tutorial de cómo instalar y usar Hadoop:
 - http://www.bogotobogo.com/Hadoop/BigData_hadoop_Install_on_ubuntu_single_node_cluster.php
 - http://www.bogotobogo.com/Hadoop/BigData_hadoop_Running_MapReduce_Job.php

Bibliografía: libro

- Hadoop: The Definitive Guide, 3rd Edition:
 - <http://shop.oreilly.com/product/0636920021773.do>
 - <https://github.com/tomwhite/hadoop-book/>



Bibliografía: TFG

- Extracción de información social desde Twitter y análisis mediante Hadoop.
 - Autor: Cristian Caballero Montiel
 - Tutores:
Daniel Higuero Alonso-Mardones y Juan Manuel Tirado Martín
 - <http://e-archivo.uc3m.es/handle/10016/16784>
- Adaptation, Deployment and Evaluation of a Railway Simulator in Cloud Environments
 - Autora: Silvina Caíno Lores
 - Tutor: Alberto García Fernández

Agradecimientos

- Por último pero no por ello menos importante, agradecer al personal del Laboratorio del Departamento de Informática todos los comentarios y sugerencias para esta presentación.



Sistemas Paralelos y Distribuidos

Máster en Ciencia y Tecnología Informática

Diseño de Sistemas Distribuidos

Máster en Ingeniería Informática

Curso 2021-2022

Sistemas escalables en entornos distribuidos

Alejandro Calderón Mateos,

acaldero@inf.uc3m.es

Óscar Pérez Alonso,

oscar@lab.inf.uc3m.es

Félix García Carballeira

fgcarball@inf.uc3m.es