Introducción al *big data*

PID_00264727

Xavi Font

Tiempo mínimo de dedicación recomendado: 2 horas

Universitat Oberta de Catalunya

Xavi Font

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Jordi Ayza Graells (2019)

Primera edición: marzo 2019

Av. Tibidabo, 39-43, 08035 Barcelona
Diseño: Manel Andreu
Realización editorial: Oberta UOC Publishing, SL

Índice

Introducción

En los últimos años la potencia de los ordenadores ha aumentado exponencialmente. Este fuerte incremento en el rendimiento de procesamiento ha ido en paralelo con la disponibilidad, la gestión y el almacenaje de grandes cantidades de datos. Todos estos elementos, junto con el desarrollo y avance de nuevas técnicas de *machine learning* para la identificación de patrones, han incidido en la importancia de analizar correctamente estos datos para convertirlos en conocimiento o ventaja competitiva para las organizaciones.

Si bien el nombre de *big data* fue inicialmente empleado en un artículo publicado en el 2001 por Doug Laney y titulado: "3D data management: controlling data volume, velocity and variety", la definición formal ha tenido diversas variaciones y dependiendo del contexto puede también cambiar. El concepto de volumen de datos se asocia a gran volumen de datos, y en este punto podemos preguntarnos: ¿qué magnitud representa? ¿Gb, Tb, Pb, Zb o Yb? Para empresas como Google, Facebook, Twitter u organizaciones como la NSA, su unidad de medida será la de Zb o Yb. Para empresas medianas, quizá estas magnitudes son desconocidas, pero sí que pueden tratar problemas de *big data*, y en su caso su unidad de medida estará alrededor de los Tb.

Otro aspecto importante son las soluciones cada vez más próximas a las necesidades de los usuarios. La irrupción de la sensorización y el acceso a estos volumenes de datos ha dado lugar a otro paradigma, pasando de los sistemas basados en RDBMS a sistemas No SQL. Ejemplos que podemos destacar son las soluciones *open source* Apache con la solución de Cassandra.

Existen una serie de herramientas asociadas al *big data*, como Hadoop, MapReduce, Spark, Pig, Hive y Apache Cassandra, que constituyen la base de muchas soluciones software. Ninguna de estas herramientas será tratada en este módulo con detenimiento, pero alguna de las propuestas se describirán para poder evaluarlas como posibles alternativas reales de implantación.

Objectivos

El principal objetivo es entender la importancia y las implicaciones a las que hace referencia el término *big data*. En concreto los objetivos son los siguientes:

1. Entender la definición conceptual del *big data* y el porqué de su aparición y relevancia.
2. Conocer arquitecturas *big data*. En concreto, conocer alguna de las propuestas que tenemos a nuestra disposición. En ocasiones podemos derivar toda la infraestructura a un entorno de *cloud computing*, si desde la perspectiva de nuestro negocio es beneficioso.
3. Conocer alguna solución que el mercado pone a nuestra disposición.
4. Saber qué es un problema de *big data* y qué no es un problema de *big data*.

1. Entorno y contexto para el crecimiento y la importancia del *big data*

Imaginemos nuestra calle no únicamente como el lugar donde residimos sino como un sistema. ¿Quién genera información relativa a nuestra calle? Hay una lista considerable de agentes externos que recogen datos de lo que pasa en nuestra calle: un grupo de datos son los que nos generan dolores de cabeza (por representar pagos por servicios que hemos contratado):

- Uso del teléfono, móvil e internet
- Hábitos de consumo eléctrico
- Utilización de gas
- Pago impuestos municipales-catastro
- Consumo de agua

Otro grupo de datos que en ocasiones desconocemos pero que también inciden en lo que pasa en nuestra calle (sistema):

- Tráfico
- Información meteorológica: temperatura, humedad, presión atmosférica
- Información de calidad del aire (estaciones terrestres) CO_2 $PM10$, entre otros
- Datos provenientes de satélites de observación de la tierra
- Estadísticas oficiales: Eurostat, INE, Idescat, etc.
- Hábitos de consumo (tarjetas de crédito)
- Interacción en redes sociales
- Otros

Todo este conjunto masivo de datos, que tiene como elemento común nuestra calle, configura un escenario típico de *big data*. En este escenario podemos desgranar algunas características interesantes y que pueden suponer diferentes grados de complejidad.

Primero: gran cantidad de datos que se generan de forma permanente y por unidad de tiempo (cada segundo 1Tbyte).

Segundo: gran variedad de formatos y tipos de datos. Podemos tener temperatura en grados centígrados, imágenes provenientes de Meteosat, texto escrito en un foro, mensajes de aplicaciones móviles tipo WhatsApp o enlaces a páginas web (URL).

Tercero: los datos se generan de manera instantánea y podemos pensar que estamos en un entorno en tiempo real.

En la actualidad, las tendencias hacia una revolución digital en los procesos de manufacturación, conocidas como industria 4.0, se sustentan en diferentes tecnologías: IoT, CPS, *cloud computing* y BDA. Brevemente, el concepto IoT se refiere a un mundo interconectado con el propósito de recopilar e intercambiar datos. En general, IoT es capaz de ofrecer conectividad avanzada de sistemas físicos, sistemas y servicios. El acrónimo CPS se refiere a sistemas ciberfísicos (en inglés, *cyber-physical system*), que no es más que un mecanismo a través del cual objetos físicos y software están íntimamente imbricados, facilitando de este modo el intercambio de información. El concepto de *cloud computing* corresponde al uso de servicios computacionales y escalables sobre internet. El último acrónimo es *big data analytics*, e intenta resolver de qué forma se pueden procesar los datos con la intención y el propósito de dar la información correcta para el objetivo correcto en el momento adecuado.

2. La definición del *big data*

El concepto de *big data* se inició como una innovación tecnológica en el ámbito de la computación distribuida para tratar con los grandes volúmenes de datos que inundan el día a día de cualquier negocio. Google recibe cada día 3.500.000.000 de búsquedas. Lo realmente importante es el provecho que obtenemos de los datos almacenados, y en este sentido cómo analizamos y visualizamos los datos para realizar mejores decisiones al tiempo que ayudamos a dirigir y definir las estrategias de negocio exitosas.

Las tres V han sido extendidas y ampliadas por otros investigadores. De este modo, podemos encontrar la V de veracidad (introducida por IBM), o por ejemplo variabilidad o valor. Generalmente, la descripción asociada al *big data* se vincula con las limitaciones de las infraestructuras tecnológicas, con la capacidad de procesamiento y almacenamiento de estos datos.

2.1. Volumen

La capacidad de almacenamiento en sistemas tradicionales se ve desbordada por el incremento exponencial de los datos. Podemos pensar en los entornos donde hay un sistema gestor de bases de datos relacional (SGBD) que no estaba diseñado para soportar estas cargas de datos ni su variabilidad.

Como ya hemos mencionado, el valor del volumen que de forma objetiva identifica un proyecto de *big data* no está escrito y depende de la naturaleza de nuestra infraestructura tecnológica actual. Estos valores pueden moverse por encima de los petabytes.

2.2. Velocidad

El concepto de velocidad tiene dos vertientes que deben considerarse. La primera tiene relación directa con el crecimiento de los datos generados. Si cada segundo generamos a través de nuestros sensores un terabyte, esta velocidad de carga es muy superior a imaginar una entrada de datos del orden de megabytes. Adicionalmente, deberemos dar respuesta (en muchos casos en tiempo real) a unas necesidades de información y, por tanto, procesamiento de los datos de manera eficiente.

2.3. Variedad

En la descripción de esta V hay que entender la procedencia y diversidad de los datos generados a través de los diferentes sensores y/o procesos de interés. Muchos datos provienen de un *streaming*, es decir, una cámara o un micrófono envía imágenes o audio sin que exista nada más que la imagen en algún formato predeterminado (.jpeg, .giff) o un audio (.mp3), pero no son datos estructurados. El análisis de datos no estructurados como estos, o por ejemplo el texto, incrementa la dificultad de tratamiento.

En otras ocasiones los datos son elementales (no están compuestos) y presentan una ventaja evidente en su procesamiento. Ejemplos de datos estructurados son la temperatura de un motor o diversos parámetros relacionados con el rendimiento de una línea de producción.

Existe una tipología de datos que están a medio camino entre los estructurados y los no estructurados. Esta tipología, llamada formalmente datos semiestructurados, contiene información de cómo interpretar los datos contenidos en el documento. Podemos encontrar ejemplos en los formatos HTML, JSON o XML.

2.4. Veracidad

Una gran cantidad de datos a nuestro alcance no representan absolutamente nada. La cantidad no se traduce en valor de ningún modo. Es importante determinar la exactitud y validez de los datos para garantizar que cualquier procedimiento de análisis que pretenda identificar patrones y conocimiento estratégico tenga garantías de representar la realidad de manera precisa.

Nuestras interpretaciones se verán comprometidas si la calidad de los datos no está garantizada.

2.5. Interpretación de las cuatro V

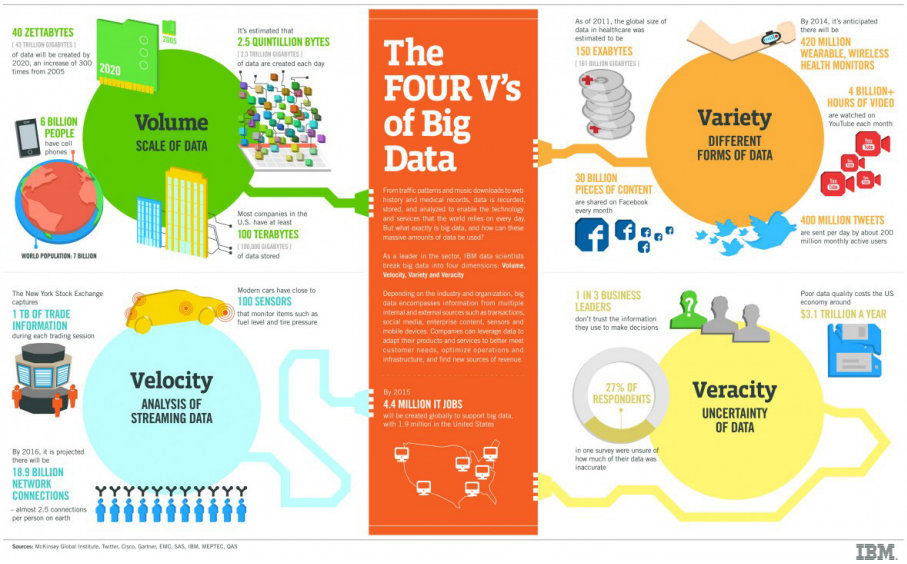
Muchas son las empresas que ofrecen infografías para explicar estos conceptos involucrados en la definición del *big data*, como la que se muestra en la figura 1.

En ocasiones se añaden otras características que permiten ofrecer nuevas formas más precisas de definir un entorno *big data*. En el caso, de nuevo de IBM añade las siguientes uves:

- V de *visibility* (visibilidad), que es un concepto que va ligado a la nueva ola de soluciones administrativas (y de organizaciones privadas) que ofrecen los datos al resto de los usuarios. Los llamaríamos entornos *open data*.

- V de *values* (valor en los datos). Esto indica que tenemos una gran variedad y tipología de valor en los datos.

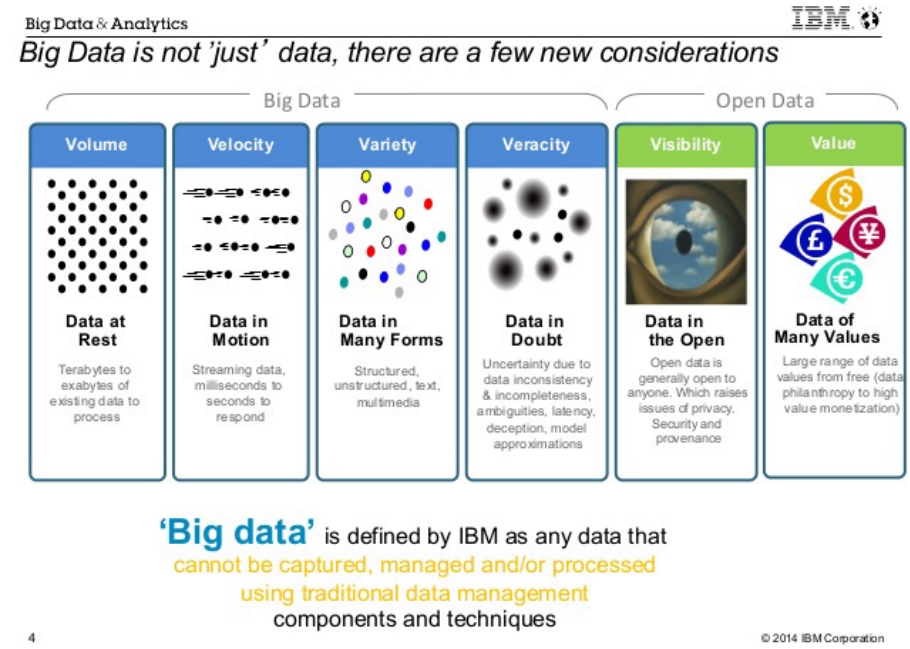
Figura 1. Definiciones propuestas por IBM



Fuente: https://www.ibmbigdatahub.com/infographic/four-vs-big-data

En la figura 2 se visualizan de nuevo los conceptos presentados relativos a las 4 V (volumen, velocidad, variedad y veracidad), pero se añaden dos conceptos más. Estos dos últimos términos corresponden a visibilidad y valor. Visibilidad hace referencia a los conceptos de *open data* y el de valor, al concepto estratégico de los datos a nuestra disposición.

Figura 2. Nuevas consideraciones que añadir a las 4 V. IBM añade las V relativas al concepto de *open data*



Fuente: https://www.slideshare.net/AndersQuitzauIbm/big-data-analyticsin-energy-utilities

La aparición de este nuevo contexto va a permitir a las empresas abordar problemas y generar conocimiento estratégico que anteriormente eran impensables. La publicidad y el éxito de los métodos analíticos se deben en parte a este tipo de aproximaciones (ver la tabla comparativa en la figura 3).

Figura 3. Evolución de los enfoques

Transaction System	Business Intelligence System	Business Analytics System	
Automate Process	Support Decision	Enhance and discover Decision	✓
Designed for Efficiency	Designed for Effectiveness	Designed for Prediction and for Gaining Insights	✓
Structure the Business	Adapt to the Business	Drives the Business	✓
React to the Event	Anticipate to the Event	Discover new Events	✓
Optimized for Transactions	Optimized for Queries	Optimized for Visualization and Machine Learning	✓
Transactional Data	Business Intelligence Data	Business Analytics Data	✓

La complejidad de los entornos *big data* puede generar dudas. Desde la perspectiva empresarial, quizá, pueda sostenerse el no actualizar los sistemas de información al cubrir gran parte de las demandas actuales.

¿Qué motivos pueden justificar la cautela a la hora de implementar soluciones *big data*?

- Insuficiente comprensión y aceptación del *big data*: se desconoce el concepto, los beneficios que aporta, la infraestructura necesaria, el ROI.

- Confusión entre las soluciones y tecnologías disponibles: aparecen nombres como Cassandra, Apache Hadoop, MapReduce y Apache Spark, entre otros.

- Coste de las soluciones *big data*: se tiende a pensar que los costes asociados son excesivamente caros, pero la realidad es diferente, ya que tenemos soluciones como Apache Cassandra que son soluciones *open source* (gratuitas) y en todo caso la empresa debe valorar los costes relativos a la infraestructura. Las opciones de *cloud* son una alternativa que considerar.

- Complejidad en la gestión de la calidad de los datos: la incorporación de grandes volúmenes de datos tiene sentido si somos capaces de sacarle provecho y de garantizar la calidad de estos. Hay soluciones para resolver este tipo de problemas.

- Problemas de seguridad: punto especialmente delicado pero que puede tener especial relevancia en entornos donde los datos tienen un alto nivel de confidencialidad. De qué riesgos somos conscientes:

  1) falsa generación de datos,

  2) protección criptográfica,

  3) acceso a información sensible gracias al uso de ML,

  4) gestión del control de acceso en función de usuario y tipo de información que puede manejar,

  5) garantizar la calidad y la procedencia de datos,

  6) desconocimiento de las bases de datos NoSQL y falta de enfoque de seguridad con ausencias de auditorías en esta materia.

3. Arquitectura *big data*

Las plataformas y las infraestructuras que dan soporte a proyectos *big data* son en muchos casos complejas. Podríamos añadir que pueden aparecer importantes aspectos no incluidos en la definición del concepto de *big data*, pero que pueden tener una criticidad o importancia extrema (i.e. seguridad).

Desde una perspectiva conceptual, podemos entender el problema de la arquitectura definida por proyectos *big data* como uno asociado a los sistemas de computación distribuidos (DCS: *distributed computer system*). Estos sistemas se componen de diversos elementos software y están en múltiples máquinas pero se ejecutan como un único sistema.

3.1. Apache Hadoop

Podemos verificar la definición que Apache Hadoop hace de su solución:

> «The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing.»

Enlace de interés

Toda la información actualizada de Apache Hadoop puede encontrarse en: https://hadoop.apache.org/

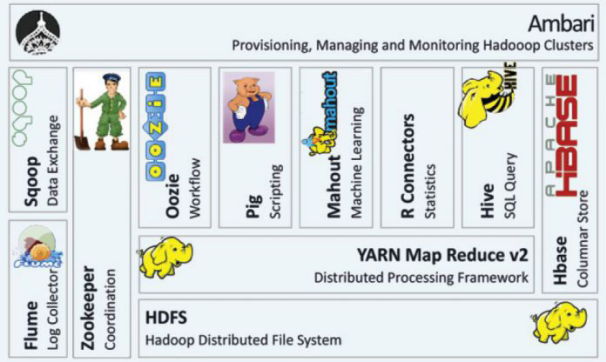
Utizado inicialmente por Yahoo! y Facebook, Hadoop es una plataforma de procesado de datos *open source* que permite almacenar y procesar grandes cantidades de datos sobre un conjunto de clústeres. Los elementos más significativos de esta arquitectura son el sistema de ficheros distribuido (HDFS, *Hadoop distributed file system*) y el modelo de programación MapReduce.

MapReduce

MapReduce es un paradigma de programación para procesar y manejar grandes conjuntos de datos.

A pesar de sus muchas ventajas, tiene algunos elementos mejorables relativos a la seguridad, su falta de ajuste a datos de menor tamaño y sus altos *overheads* en operaciones de entrada y salida. En la figura 4 se aprecia el ecosistema de Hadoop.

Figura 4. Hadoop-Echosystem (https://hadoop.apache.org/)



Fuente: https://www.quora.com/What-is-a-Hadoop-ecosystem

3.2. Cloudera Data Hub

Esta solución introduce Cloudera Enterprise Data Hub como un producto destinado a convertir los datos en propuestas de valor para el negocio. Ofrece un *framework* basado en Hadoop para entornos IoT y de *big data*.

¿Qué es Cloudera? Cloudera ofrece una plataforma escalable, flexible e integrada que facilita la gestión de grandes volúmenes de datos, caracterizados por un rápido crecimiento y sujetos a una gran variedad de tipos de datos. Los productos y soluciones de Cloudera permiten implementar y administrar Apache Hadoop y los proyectos relacionados, de manera que facilitan la manipulación y análisis de los datos (garantizando su seguridad y protección).

Cloudera proporciona los siguientes productos y herramientas:

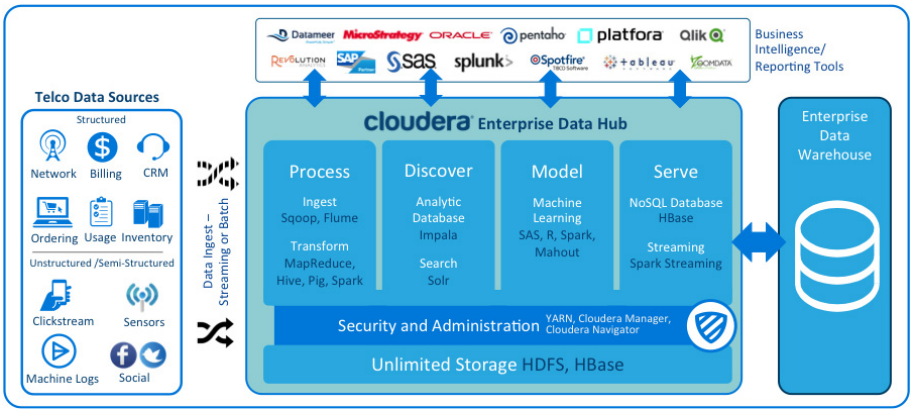
- **CDH**: Cloudera Distribution of Apache Hadoop.
- **Cloudera Manager**: aplicación utilizada para implementar, administrar, monitorear y diagnosticar problemas con el despliegue de la distribución Cloudera de Apache Hadoop (CDH).
- **Cloudera Navigator**: herramienta de seguridad y administración de datos para la plataforma CDH.

Las características de CDH (desde la perspectiva de Cloudera):

- **Flexibilidad**: permite que se almacene cualquier tipo de datos y garantiza la manipulación de estos. Ofrece variedad de marcos de cómputo, como por ejemplo procesamiento por lotes, SQL interactivo, búsquedas, aprendizaje automático y cálculo estadístico.
- **Integración**: permite adaptarse a su sistema con una plataforma completa de Hadoop que funciona con una amplia gama de hardware y soluciones de software.
- **Seguridad**: permite procesar y controlar datos sensibles.
- **Escalabilidad**: permite escalar y ampliar para satisfacer necesidades.
- **Alta disponibilidad**: permite operaciones de misión crítica.
- **Compatibilidad**: se adapta a su actual infraestructura.

La figura 5 muestra la arquitectura de Cloudera implementada con éxito en la empresa Telcos.

Figura 5. Cloudera Enterprise Data Hub



Fuente: http://vision.cloudera.com/driving-value-for-telcos-with-an-enterprise-data-hub/

Enlace de interés

La página de Cloudera ofrece multitud de documentación e información de interés: https://www.cloudera.com/

3.3. Cassandra

Apache Cassandra es una base de datos NoSQL ideal para entornos de alto rendimiento y de alta velocidad. Es una base de datos distribuida, altamente escalable, diseñada para manejar grandes cantidades de datos en muchos servidores, proporcionando una alta disponibilidad sin un punto único de error o fallo.

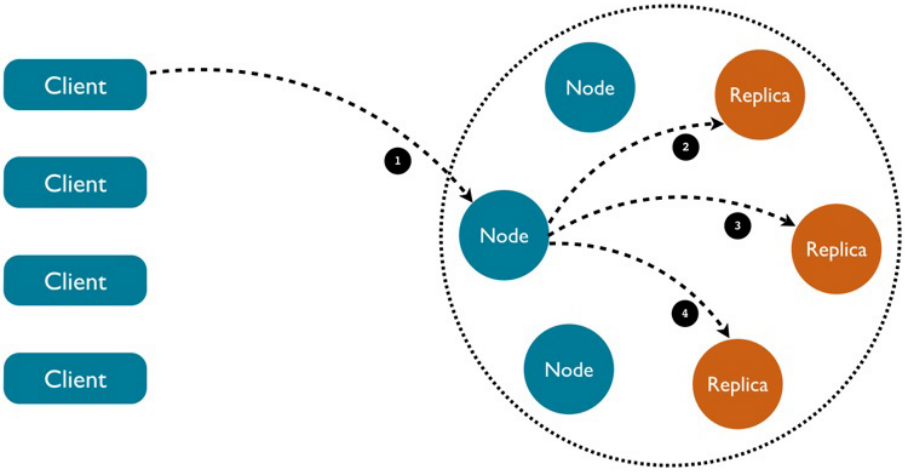
De este modo, podemos entender que la arquitectura de Cassandra se basa principalmente en garantizar que el sistema funcione sin fallos y en un entorno distribuido. La escalabilidad de este tipo de soluciones permite tratar entornos con volúmenes de datos del orden de petabytes y garantizar la absoluta respuesta del sistema.

3.3.1. Distribución y replicación

La arquitectura de Cassandra se basa en nodos que forman parte de un anillo, también llamado clúster. Los datos se particionan a través de los nodos y realiza las replicaciones y las copias redundantes necesarias para que si un nodo deja de funcionar, otro pueda responder de forma inmediata y garantizar que el sistema siempre está disponible.

El sistema garantiza que al hacer una operación esta se replique de forma transparente al usuario. En la figura 6 se aprecia el funcionamiento de un clúster frente a la petición de un cliente. Esta replicación garantiza que en casos de caída (fallo) de algún nodo otro pueda garantizar que dispone de la información y, por tanto, responder a la petición del usuario.

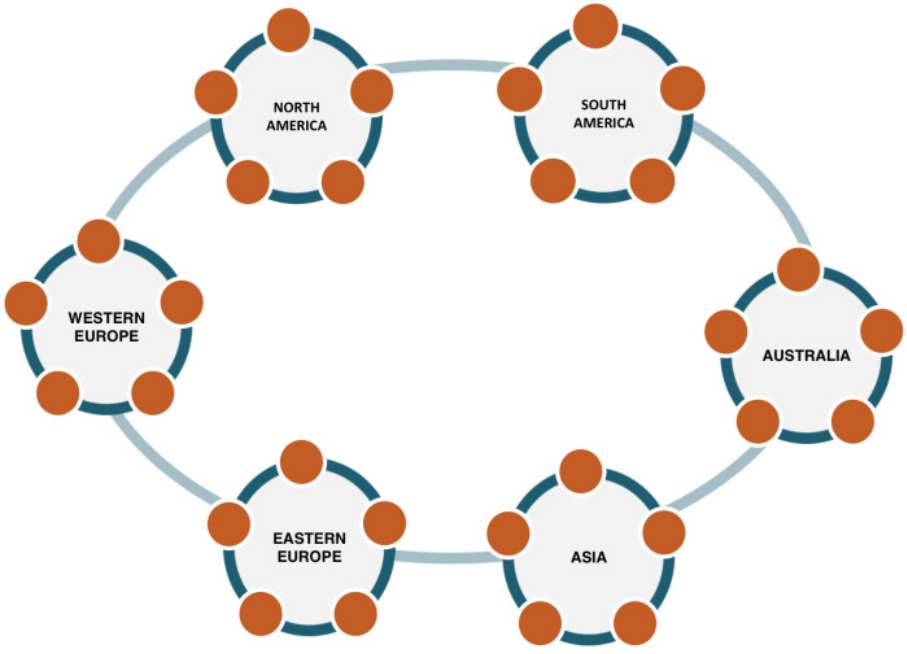
Figura 6. Diagrama de respuesta de un clúster Cassandra a una petición



Fuente: https://www.datastax.com/dev/blog/cassandra-error-handling-done-right

En la figura 7 se aprecia la verdadera potencia de la solución en un entorno de diferentes centros de datos. Cada uno de estos centros de datos está ubicado geográficamente en diferentes zonas. Cassandra va a garantizar siempre las lecturas y escrituras en el sistema bajo cualquier situación.

Figura 7. Aproximación Multi Data Center de Cassandra. Se pueden observar diferentes clústeres (o anillos) y los diferentes nodos que dan servicio en cada uno de estos clústeres



Fuente: https://www.datastax.com/dev

Este tipo de arquitectura permite una disponibilidad absoluta y ofrece un rendimiento muy alto bajo cualquier carga.

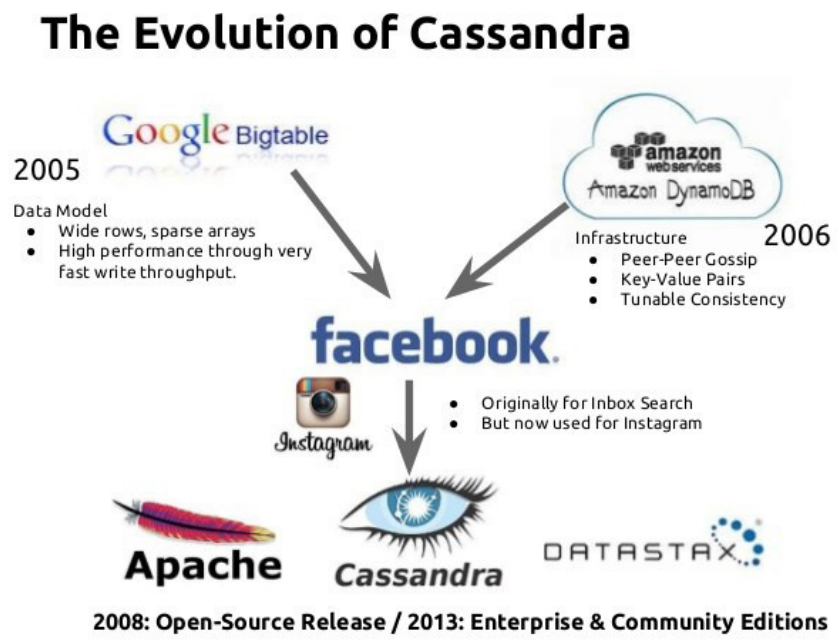
4. Soluciones *big data*

En este apartado se va a tratar con un poco más de detenimiento la solución más implementada en la actualidad. Si bien hay otras alternativas, desde un punto de vistas pedagógico entendemos que el modelo que hay detrás de Cassandra es el más atractivo y donde se pueden encontrar más referencias de éxito en los entornos reales de *big data*.

4.1. Apache Cassandra. Antecedentes

Apache Cassandra es un sistema de gestión de bases de datos NoSQL, de código abierto, distribuido y gratuito, diseñado para manejar grandes cantidades de datos en muchos servidores, proporcionando una alta disponibilidad sin ningún punto de falla, lo que garantiza siempre máximo rendimiento. Pero ¿cuáles son los antecedentes o los inicios de Cassandra?

Figura 8. Cassandra proyecto *open source* de Facebook



Fuente: http://qindasnanda.blogspot.com/2017/01/apache-cassandra-cassandra-tutorial.html

Cassandra fue un sistema diseñado inicialmente por dos exempleados de dos empresas de referencia en el ámbito tecnológico: Amazon y Microsoft. Se basaron en la solución Dynamo, una innovadora base de datos distribuida para utilizarla como plataforma por Facebook.

En la actualidad, Cassandra es mucho más que un nuevo sistema que da gran potencia a las búsquedas de Facebook. Se ha convertido en la solución seleccionada por muchas empresas por el alto rendimiento en el procesado de transacciones. Este proyecto desarrollado como *open source* está integrado en Apache y ofrece un sistema de almacenaje distribuido muy potente.

4.2. Cassandra frente a RDBMS

Como Henry Ford dijo en una ocasión:

> «si hubiera preguntado a la gente qué quería, me hubieran respondido que caballos más rápidos»

Cassandra es un sistema que dista profundamente de un sistema de base de datos tradicional. Desde un contexto histórico, IBM inventó el sistema de gestión de información en el año 1966 (IMS, *information management system*). Otro punto importante y relevante fue la aparición de los sistemas de bases de datos relacionales. También podemos destacar la publicación de Edgar F Codd (IBM): "A relational Model of Data For Large Shared Data Banks", un artículo que dio impulso a todos los sistemas que han ocupado todo el espacio tecnológico hasta estos últimos años (RDBMS, *relational data base management system*). La disrupción de nuevo de un sistema completamente distinto (NoSQL) se asemeja a la disyuntiva de H. Ford coche/caballo (obsérvese que esta comparación se hace desde la hipótesis de entornos *big data* e IoT).

Cassandra soluciona problemas de escalado, incremento del uso del sistema, el tiempo de espera en las operaciones de *join* o el de cargas masivas (imaginad los casos de Facebook, Instagram o Twiter). La idea consiste en duplicar nuestros datos con el objetivo de aumentar velocidad y ofrecer tolerancia a fallos, pero que evidentemente es un proceso contrario a los sistemas RDBMS (proceso llamado desnormalización).

La respuesta a la cuestión de si debemos implantar Cassandra dependerá de las características y el uso que demos a nuestro sistema. Si no hay problemas de escalabilidad, ni de respuesta del sistema a consultas o cargas, entonces un enfoque tradicional puede ser una alternativa razonable.

Enlace de interés

Sobre las características de un sistema RDBMS, podéis consultar información en Oracle: https://bit.ly/2DqD9aD

4.2.1. Definición de Cassandra

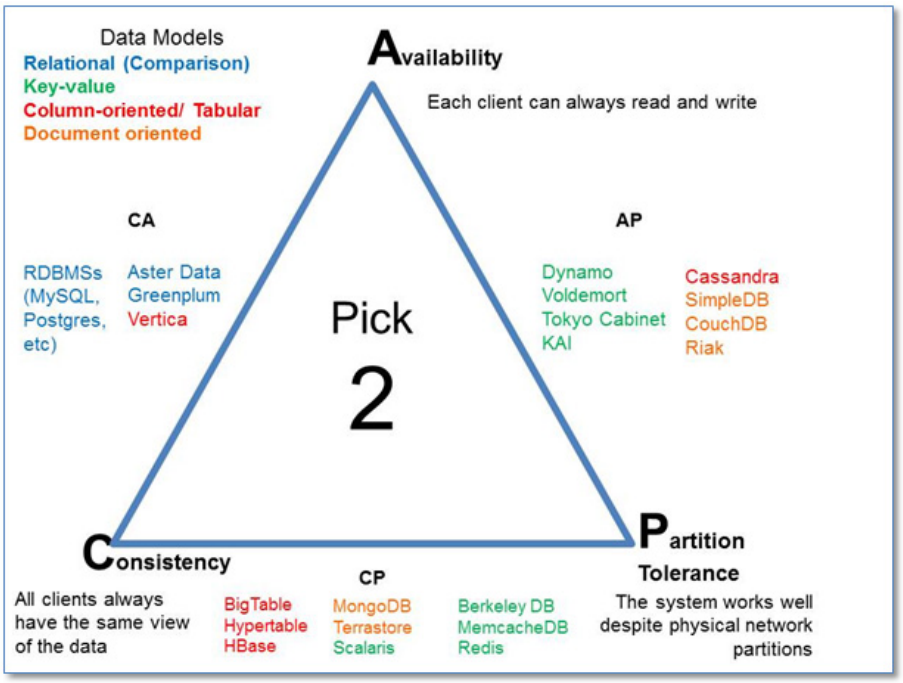
Cassandra se puede definir de una forma breve y sintetizada de la siguiente manera: es un sistema *open source*, distribuido, descentralizado, escalable elásticamente, altamente disponible, tolerante a fallos, consistente (con *tunning*), base de datos orientada a columnas que basa su diseño distribuido en Dynamo de Amazon y el modelo de datos de BigTable de Google.

Creado en Facebook, es en la actualidad usado por los portales más populares.

Analicemos los diferentes términos que ayudan a entender la potencia de Cassandra.

- **Distribuido**: capaz de correr en diferentes máquinas.
- **Descentralizado**: cada nodo es idéntico (*server symmetry*). Observad que los sistemas tradicionales tienen el par máster y esclavo, lo que dificulta su uso frente a los sistemas descentralizados.
- **Distribuido y descentralizado**: no hay posibilidad de error y además conduce a sistemas de una alta disponibilidad y con una facilidad de mantenimiento por tener todos los nodos iguales.
- **Escalable elásticamente**: el sistema podrá soportar incrementos de carga y uso sin que el rendimiento se vea afectado de manera brusca o notable.
- **Alta disponibilidad**: permite aceptar todas las peticiones que recibe sin que se produzcan pérdidas o errores o paro del sistema.
- **Tolerancia a fallos**: concepto importante porque garantiza que en caso de desastre (errores internos o externos) el sistema puede gestionarlo por la capacidad distribuida de Cassandra.
- **Consistencia**: concepto que hace referencia a que al solicitar un valor (lectura), este es devuelto entendiendo que es el más reciente (última escritura). El añadido de "tuneable" o de puesta a punto indica que se puede decidir el nivel de consistencia deseado (a costa de disponibilidad). En aplicaciones tipo redes sociales se la denomina consistencia eventual porque los datos no son críticos (el *post* que colgamos en Twitter, la foto en Facebook o Instagram). De esta forma podemos encontrar consistencia estricta, consistencia casual o consistencia eventual.

Figura 9. capModel



Fuente: https://www.mysoftkey.com/architecture/understanding-of-cap-theorem/

Teorema de CAP

Al considerar la consistencia, la disponibilidad y la tolerancia a particiones, un sistema únicamente puede conseguir dos de estos objetivos en un sistema cualquiera distribuido (teorema de CAP, véase la figura 9).

- ***Consistency*** **(consistencia)**: todos los clientes de la bases de datos leerán el mismo valor por la misma consulta realizada (incluso en actualizaciones concurrentes). En otras palabras, garantiza que cada nodo devuelva la misma escritura (la más reciente).

- ***Availability*** **(disponibilidad)**: todos los clientes de la base de datos podrán leer y escribir siempre. Se entiende que cada nodo que no falla devuelve una respuesta para todas las peticiones de lectura y escritura que recibe en un tiempo razonable. Cada nodo quiere decir en cualquier lugar de la partición de red.

- ***Partition tolerance*** **(tolerancia a particiones)**: si la base de datos se divide en múltiples máquinas, puede continuar funcionando si hay roturas en la segmentación de la red.

Debemos entender este teorema como una herramienta que ayuda a los responsables de los sistemas de información (CTO) a considerar el balance al elegir entre sistemas de datos distribuidos en red.

Una posible lectura de lo que significa cada par:

- CA (*consistent* + *available*): sistema consistente y disponible en ausencia de cualquier partición de red. Por tanto, implica que el sistema se bloqueará si hay una partición de red.
- CP (*consistent* + *partition tolerant*): sistema consistente y tolerante a las particiones (pero no disponible).

- AP (*available + partition tolerant*): sistemas que están disponibles y toleran particiones pero no pueden garantizar la consistencia. Es decir, el sistema puede devolver datos inexactos.

4.3. Características diferenciales de Cassandra

El modelo Cassandra tiene las siguientes características:

Empresas que utilizan Cassandra

¿Qué clientes valoran estas características descritas? Podemos encontrar una lista considerable de casos de implantación de Cassandra. Las empresas más conocidas que utilizan esta solución son, entre otras: Facebook, IBM, Instagram, Spotify, Netflix y Reddit.

- No utiliza SQL porque no existe lenguaje SQL (Cassandra es una base de datos No SQL).
- No tiene el concepto de integridad referencial (del modelo relacional).
- Utilización de índices secundarios.
- La ordenación no es propia. Cassandra almacena *arrays* de bytes; por tanto, no existe un *order by* y *group by* (existe un SliceRange).
- La desnormalización garantiza rendimiento.
- Diseño de patrones.
- Vistas materializadas.
- Claves agregadas.

Estructura de la BD: permite tratar cantidades masivas de datos no estructurados (no para datos transaccionales). Podemos entender que es capaz de digerir las cargas de fotografías de Instagram por encima de las 80.000.000 al día. Las familias de columnas tienen un parecido a las tablas del modelo relacional con filas y columnas (cada fila tiene una clave única). Para el acceso, Cassandra dispone de Cassandra Query Language (CQL).

Incorporación de índices: existen a partir de la versión 0.7 de Cassandra índices secundarios.

Las consultas*:

***La sintaxis es parecida a la que SQL ofrece.**

```
'SELECT * FROM customer;'

'INSERT INTO customer (custid, branch, status) VALUES('app1234', 'critical', 'A+');'

'UPDATE Customer SET branch = ''critical' WHERE oldCust > 2015;'
```

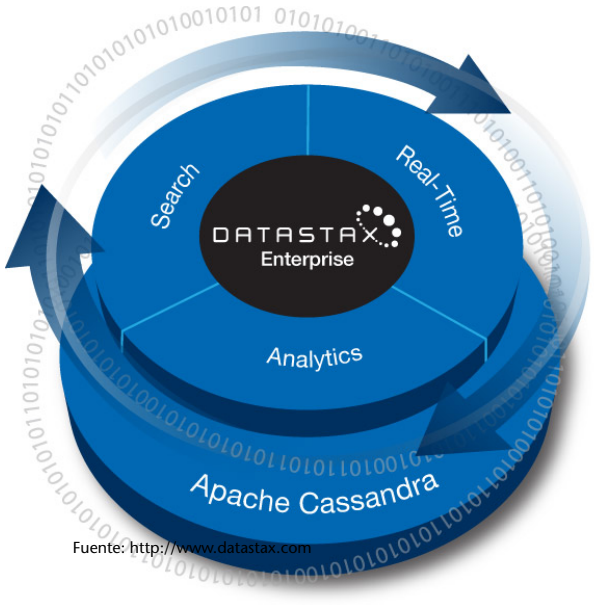
¿Cómo se ha desarrollado?: Java.

¿Qué sistemas operativos lo soportan?: Linux, OS X y Windows.

¿Quién mantiene el proyecto?: Apache Software Foundation.

¿Qué empresa está detrás de la implantación de Cassandra?: DataStax.

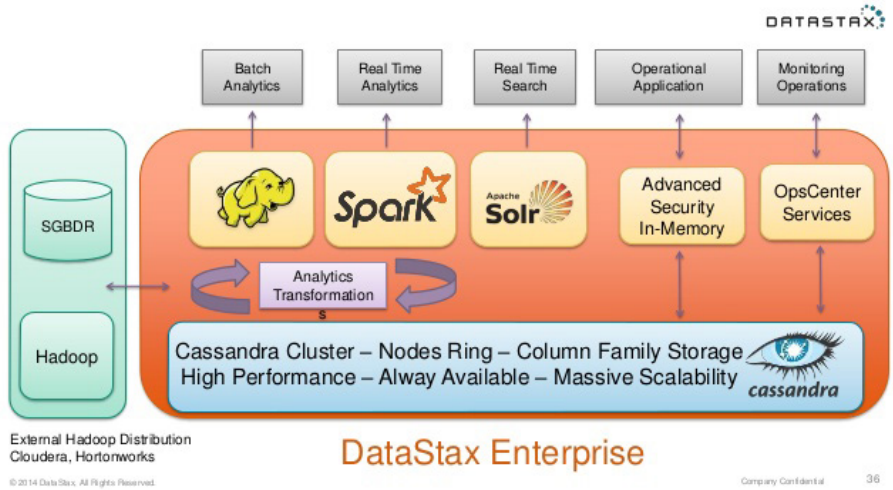
Figura 10. La empresa DataStax da servicios de consultoría para implantar soluciones *big data* con Apache Cassandra



Fuente: http://www.datastax.com

El conjunto de las soluciones que configura por ejemplo la empresa DataStax se puede ver en la figura 11.

Figura 11. DataStax Enterprise Solutions



Fuente: https://www.datastax.com

5. Caso: Plantear un entorno *big data*

Para tratar de evaluar un caso próximo, se insta a realizar un ejercicio para asentar los conceptos presentados en este módulo.

Resumen

Los puntos más relevantes que debemos resaltar son los siguientes:

- El entorno de rápido crecimiento en volumen y velocidad ofrece un nuevo concepto: *big data*. Recordad la definición de las 4 V.
- Propuestas de 2 V adicionales para enfatizar los conceptos de *open data* y del valor de los datos.
- Modelo de datos diferente para entornos *big data*. Pasamos de soluciones RDBMS (con SQL) a otro paradigma NoSQL (Cassandra).
- Las soluciones *big data* representan un reto, pero también una necesidad para garantizar la competitividad de las organizaciones y sacar el máximo provecho de la información disponible.
- Desde la perspectiva del usuario final, la parte más valorada corresponde a qué podrá realizar con su información, cómo podrá obtener conocimiento y cómo lo visualizará (figura 12).

Figura 12. CogNOS Analytics de IBM

Fuente https://www.softwareadvice.com/uk/bi/ibm-bi-profile/