Big Data Management.
Sesión 01. Introducción.

Autor: Ziani El Ali, Adil.

adilziani.wordpress.com

15 de abril de 2020

Índice

Big Data, nueva manera de gestionar datos?

Tradicionalmente, los modelos de datos se preocupaban más por gestionar los datos que explotar su valor, los datos son un activo pasivo:

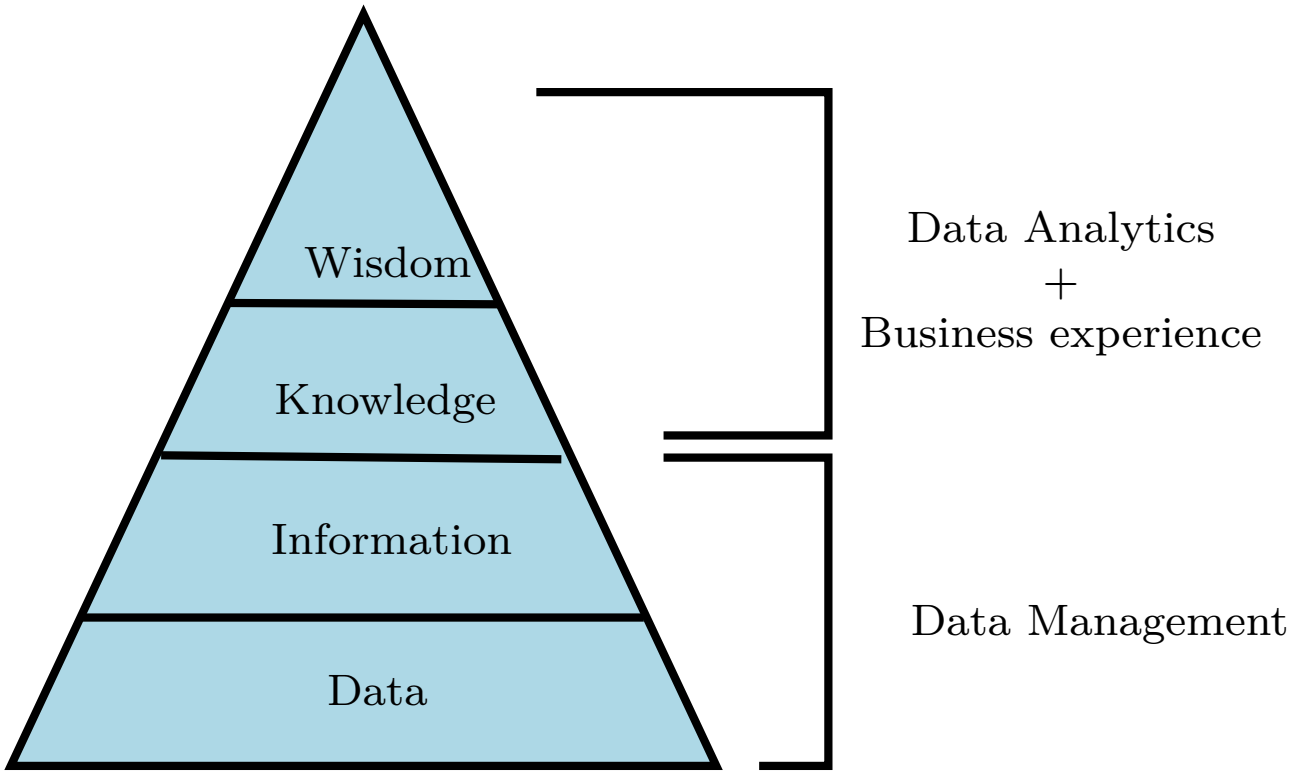
- Sistemas OLPT: datos estructurados para facilitar automatización de procesos.
- Modelos estructurados, esquema bien definida desde el principio.

Pronto surge la necesidad de explotar los datos, los datos son un activo valioso para inferir información

- Cruzar mis datos con otros datos, datos no estructurados o semi-estructurados (Schemaless), variedad de datos.
- Data Lake, primero ingestar datos y después modelar.
- Decisiones en "tiempo real", no una semana después.

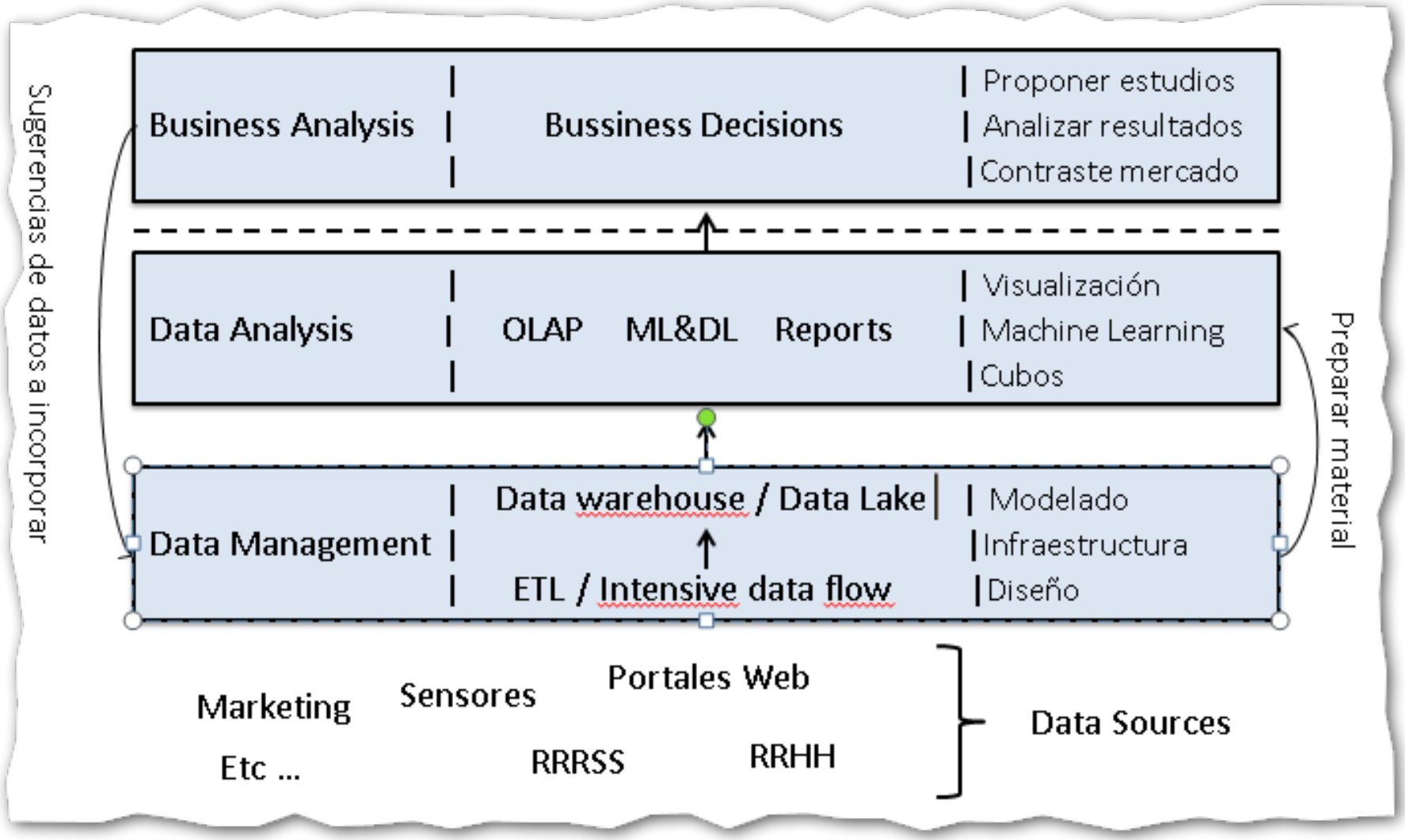
Hoy en día muchas empresas quieren
tener su Data Lake o BI system, pero en gran parte de ellas es solo un sistema de bases de datos relacionales sin una organización. "BBDD relaciones en el Lago"

Jerarquía de conocimiento



La pirámide jerarquía de conocimiento (DIKW pyramid) aplicada a los proyectos Big Data, nos muestra que el objetivo final(en la gran mayoría de los casos) es sacar información de valor al dato, todo ello en un ecosistema de Business Intelligence global.

Business Intelligence lifecycle



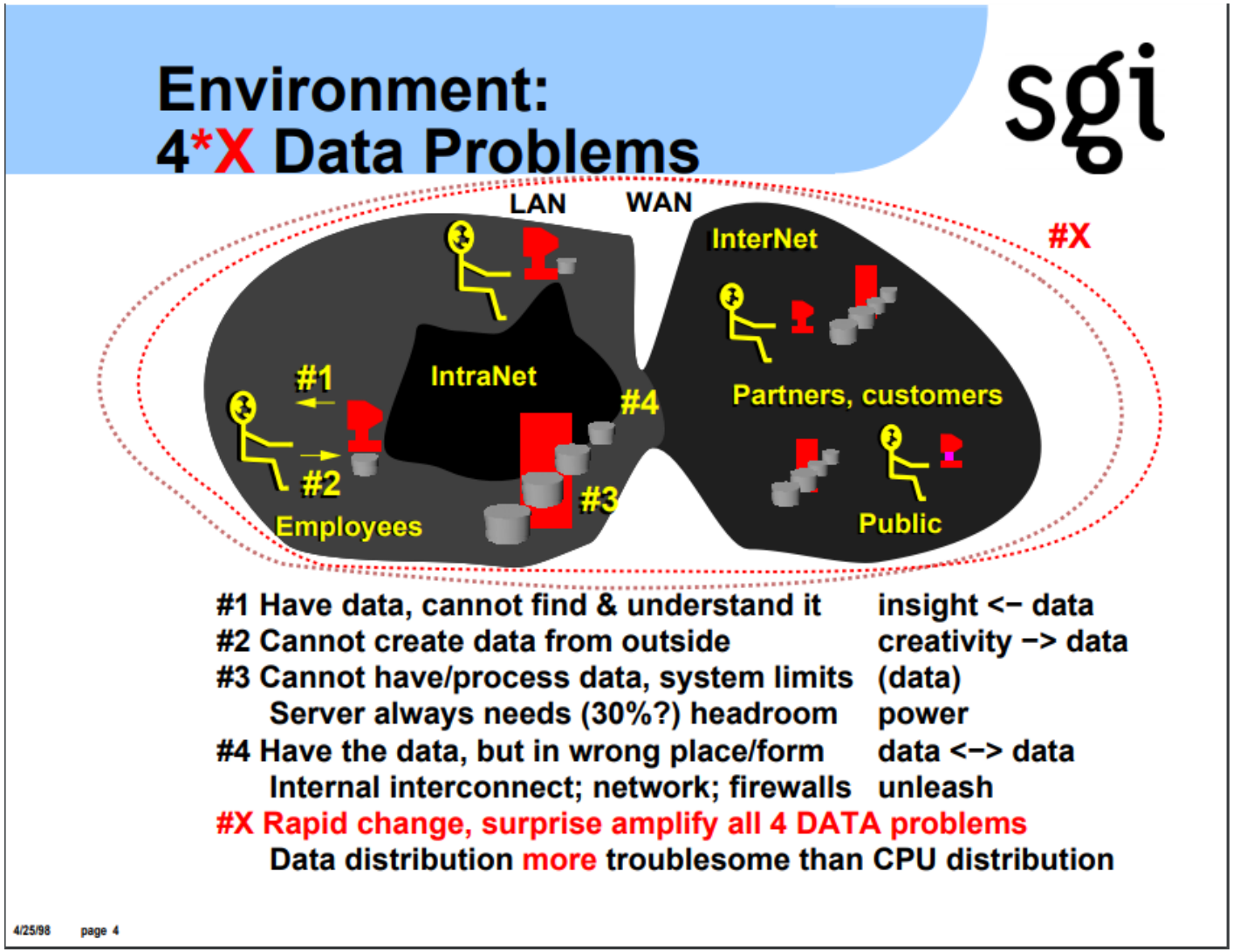


Figura 1: Diapositiva n4 de la presentación de John Mashey en 1998

Ya en 1998, John Mashey (Mashey, 1998) usaba el término Big Data para referirse a los problemas que comenzaban a surgir: más datos, crecimiento de datos a mayor velocidad, variedad de datos (fotos, videos) "Datos difíciles.

Las preocupaciones iniciales y fundamentales del Big Data se centran en dichas "3V's":

- Volumen - cada vez más cantidad de datos.
- Velocidad - los datos se generar y cambian a gran velocidad. piensa en RRSS
- Variedad - diferentes fuentes, diferentes formatos

Estas principales V's, y sobre todo las dos últimas, no son el punto fuerte de los sistemas relacionales tradicionales, a modo de ejemplos:

- Los sistemas tradicionales se basan en formas normales(poca variedad) y consistencia(IO pasa por disco mas latencia)
- Los ALTER TABLE reordenan todo, son muy costosos y por tanto el schema ha de estar bien meditado desde el inicio
- Hacer evolucionar un Warehouse tradicional es muy costoso.

No obstante los sistemas relacionales prestan alto rendimiento
en sistemas transaccionales, donde prima gestionar el dato de manera segura.

Esto no es todo, surgen **nuevas V's** que se transforman en vital importancia

Figura 2: From `https://reportedigital.com/`

Hoy en día, las principales preocupaciones son de donde obtener el dato y cuál es el dato correcto que añade valor al negocio.

- **Valor** - que el dato contenga información relevante.
- Veracidad - que el dato representa lo que de verdad se cree que representa y que sea cierto.

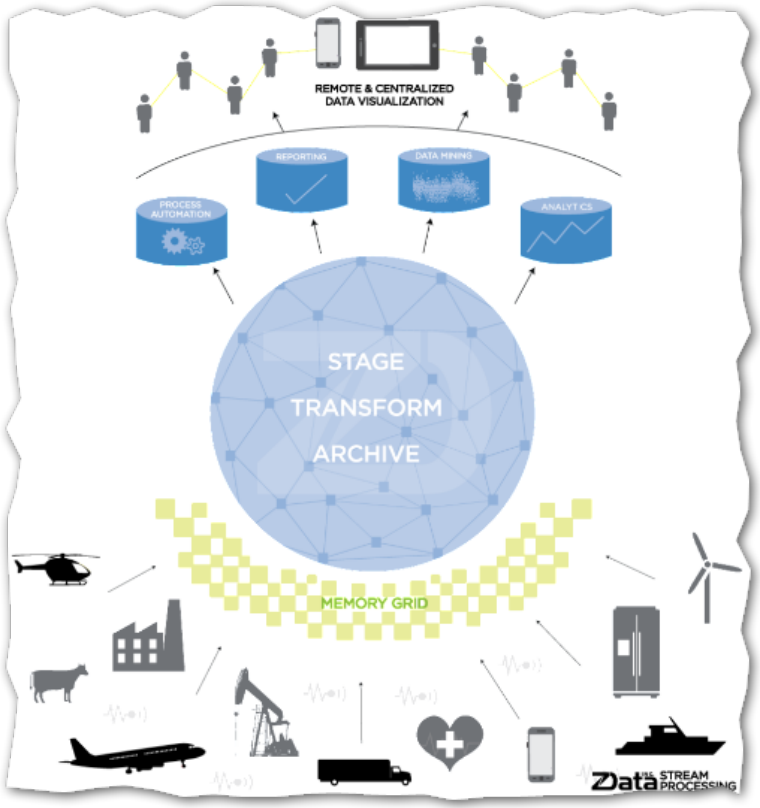
Entonces qué es el Big Data ?

Conjunto de disciplinas, herramientas, metodologías e infraestructuras para tratar y explotar el dato intentando responder a las diferentes V's.

Challenges

Responder a las diferentes V's requiere resolver diferentes problemas tanto a nivel de hardware como software. **Un nuevo ecosistema es necesario**.

- **Volume, Velocity:**
  - Data Storage (Distribución, Escalabilidad)
  - Data Replication (Disponibilidad)
  - Data Ingestion (Streams)
- **Variety:**
  - Data Modeling (Key-Value, Document-Oriented, Graph, Streams ...)
  - Data Migration (del Data Wearhouse al Data Lake)
  - Data quality
- **Value:**
  - Legislative challenges.



Challenges a la hora de poner en marcha un proyecto Big Data.

- On premise vs en Cloud ?
- Gestión y modelado. De dónde viene el dato?, cómo viene?, cómo almacenarlo?
- Calidad del dato. Proyectos de cleaning ?
- Modelado para Analysis. Qué objetivo tengo?, qué análisis fundamentales se harán ?, Qué algoritmos irían mejor?
- Gestión y coordinación de recursos humanos.

En algunas ocasiones, los proyectos Big Data atraen a las empresas con la intención de resolver problemas comerciales más rápido y añadir valor, lo que los tienta a desarrollar su solución Big Data ideal, que a menudo, acaba sin entregar nada. Un equipo coordenado y especializado, que conoce dividir el problema en fases es fundamental.

Paradigma NoSQL

En los sistemas de bases relacionales, todo el modelado es el mismo, tablas. Y las soluciones a los problemas son también similares:

- Sistemas enfocadas en escrituras?
  - Data Storage: normalización
  - Consultas: indices (B+, Hash), joins (hash join, merge join)

- Sistemas enfocadas a lecturas?
  - Data Storage: desnormalización
  - Consultas: índices (Bitmap),joins (star join), vistas materializadas.

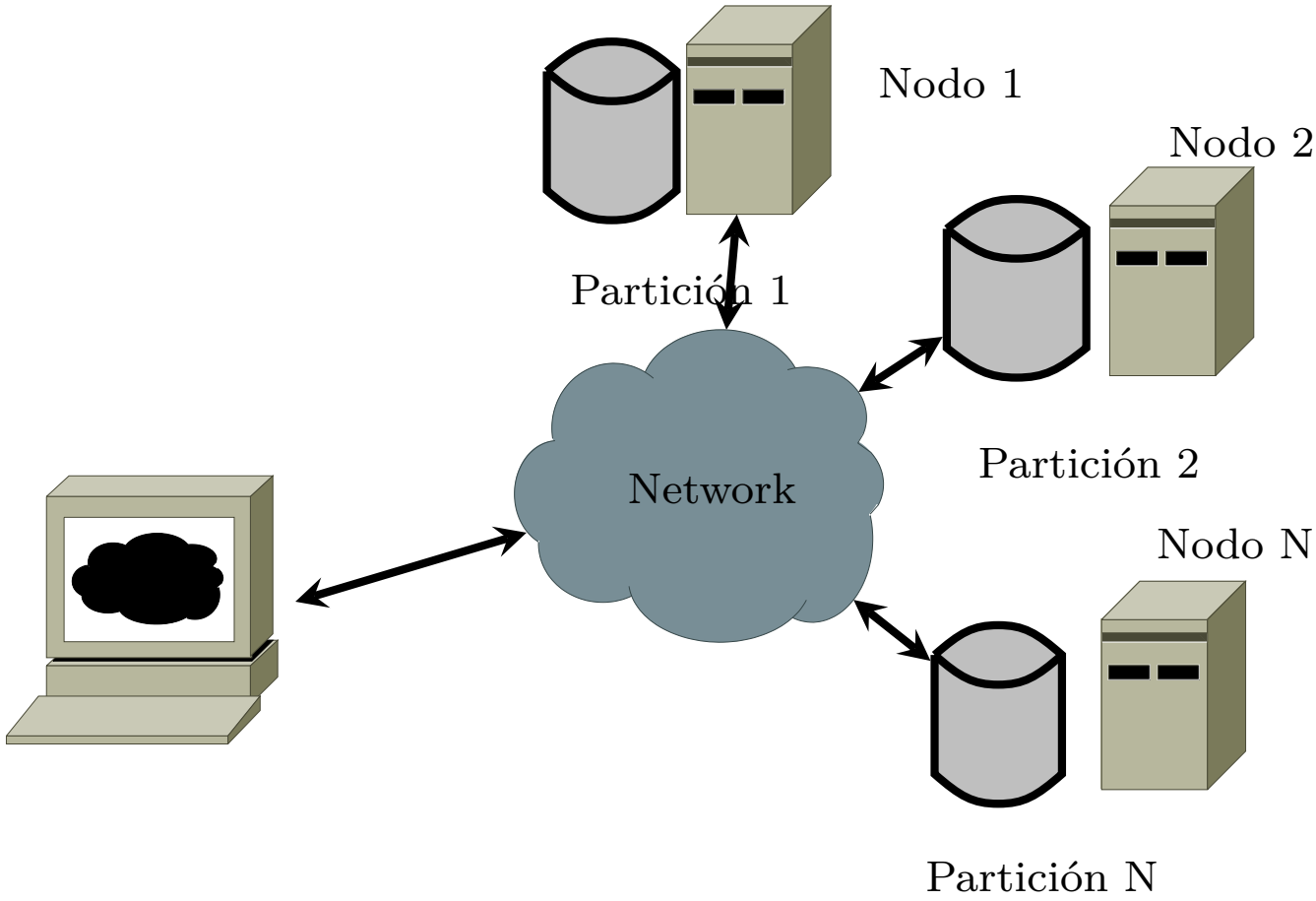
Y para sistemas con necesidad de lecturas y escrituras masivas simultáneamente?

Divide y vencerás.

Uno de los pilares del Big Data es la distribución, pues la infraestructura se basa en el multi-procesamiento con más de una máquina (nodos).

- Lecturas masivas? - Replicación de los datos.
- Escrituras masivas? - Frangmentacion de los datos.
- Procesamiento de gran volumen? - Paralelismo.

– La replicación también es parte de la solución para la disponibilidad del sistema.
– La fragmentación o particionado de los datos es también parte de la solución al procesamiento en paralelo.


Nodo 1
Nodo 2
Partición 1
Network
Partición 2
Nodo N
Partición N

NoSQL

Recordando el **objetivo de una base de datos**, éstas tienen como propósito **guardar datos y encontrarlos lo más rápido posible**. Como podemos tener proyectos de diferente índole y modelos de datos distintos, SQL y NoSQL se complementan para perseguir dichos objetivos según el modelo de datos.

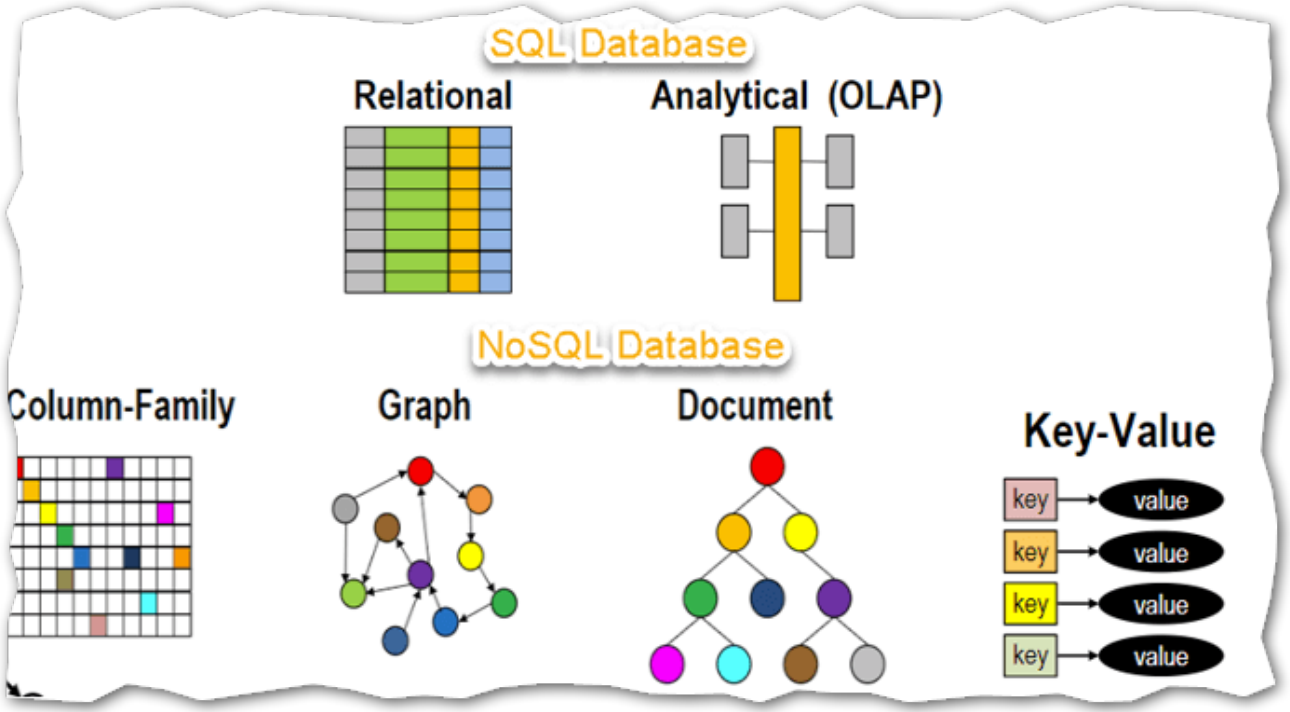
NoSQL, fue introducido por Carlo Strozzi en 1998 con un significado de Not SQL dado que su base de datos relacional no usaba SQL como lenguaje, pero en el entorno Big Data, nos referimos a NoSQL como "Not only SQL".

Las bases de datos relacionales se basan en normalizar, premiar la consistencia y coherencia de los datos por lo que ofrecen un sistema útil y fiable para sistemas transaccionales. Sin embargo, podemos tener otras necesidades que priman más una libertad en esquemas, una gestión en "real time", etc. Por lo que diferentes modelos NoSQL se necesitan considerar.

Principales características NoSQL

- Flexibilidad. Diferentes soluciones a diferentes problemas.
- Nuevas arquitecturas de datos.
- Ciertas soluciones más basadas en memoria que en disco. Más performance, menos consistencia.
- Distribución y replicación: Desnormalización.
- Fragmentación.
- Paralellismo.
- Bloom filter, evitar falsos positivos.

Modelos NoSQL



- Relacional: Oracle MySQL, IBM DB2, Teradata, PostgreSQL...
- Columnar: Vértica,...
- Graph: Neo4j, Giraph, GraphX,...
- Document oriented: MongoDB, CouchDB,...
- Key-Value: Cassandra, Hbase, HDFS,...
- Streams: Spark Streaming, Apache Flink,...

Más ejemplos de bases de datos en (Strauch, s.f.)

Cloud & On premise

Grandes empresas y gobiernos, tenían (y mantienen) una tendencia de disponer de sus propios Mainframes IBM para computar los datos (uso de Cobol, CICS y DB2 principalmente). Requiere de mucha inversión y el paradigma está enfocada a procesamiento en local[1] y los sistemas de datos son relacionales.

En los modelos Big Data, el paradigma se caracteriza por distribución y cómputo en paralelo. "Varias máquinas trabajan juntas en la misma operación", una gestión diferente a la de un Mainframe. La infraestructura en este paradigma es diferente, disponer de racks con diferentes slaves o esclavos para realizar tareas.

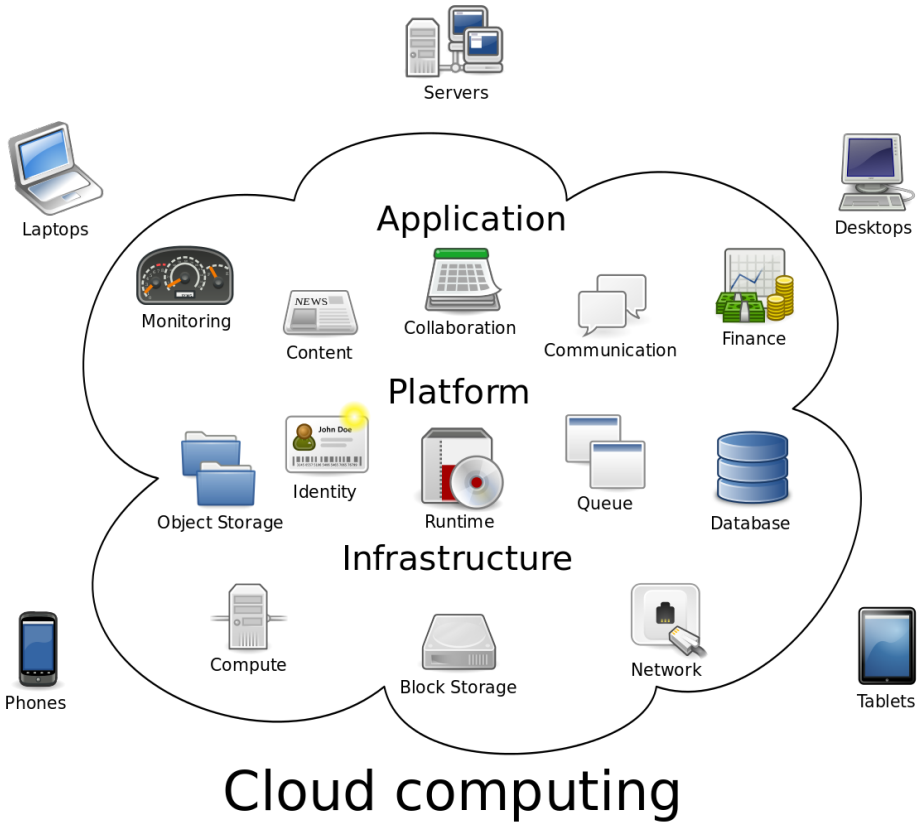
Cuántos racks y cuantos nodos necesito para mi negocio?, Siempre necesitaré todo ese potencial?

Figura 3: Racks, Mainframes

[1] También gestionan gran cantidad de datos en procesos Batch con millones de clientes

Cloud computing

El "cloud" es un paradigma en el cual, alquilamos acceso a diferentes servicios como recursos de almacenamiento, de cómputo y de gestión para montar nuestro sistema sin necesidad de disponer del hardware y las licencias necesarias.



Tipos de servicios

Infrastructure as a Service (IaaS)

- Obtiene acceso remoto a servidores vía protocolos de conexión remota VPN, SSH,etc. Por lo general este servicio cubre acceso a recursos hardware: computadoras, red, virtualización.

Platform as a Service (PaaS)

- Obtiene los módulos de software necesarios para ejecutar programas: bases de datos, servidores web, herramientas de desarrollo.

Software as a Service (SaaS)

- Obtiene acceso a software listo para usar: visualización, email, comunication.

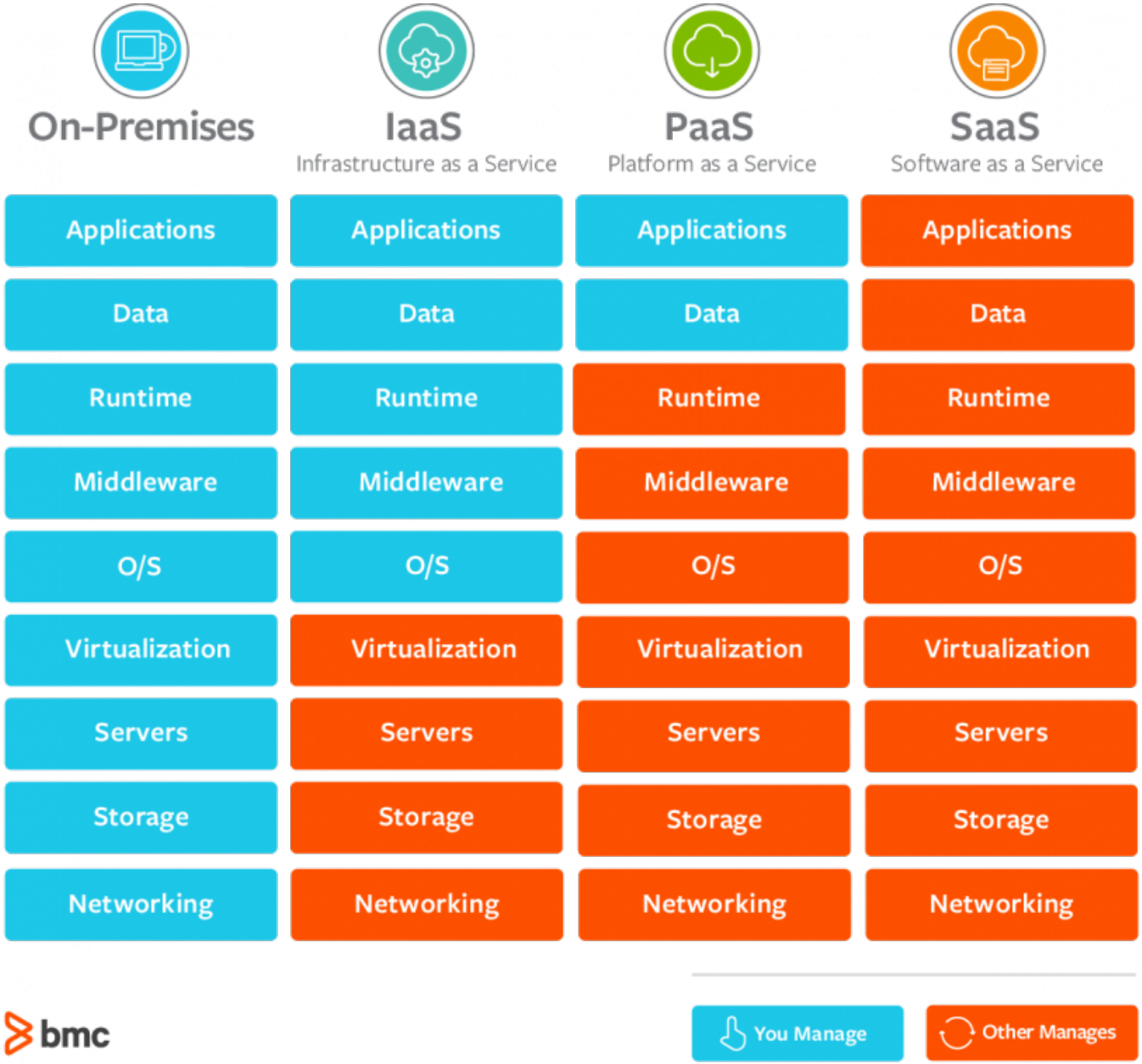


Figura 4: From `https://www.bmc.com/blogs`

Ventajas vs inconvenientes

Ventajas

- Escalabilidad y flexibilidad. Usa lo que necesitas de disco y CPU cuando lo necesitas.
- No inversión inicial en Hardware. Si el proyecto falla, se traduce en darse de baja del servicio.
- Management y configuración. El proveedor suele encargarse de actualizaciones y mantenimiento.

Inconvenientes

- Seguridad y tranquilidad sobre mis datos. Dónde se halla el CPD que aloja mis datos?, como son las normas en ese territorio?
- Dependencia del proveedor, si programa indisponibilidad temporal afecta al cliente y a su servicio.
- Si la demanda aumenta sobre el proveedor pueden aumentar los precios.

Algunos proveedores

- Amazon Web Services
- Google Cloud
- Microsoft Azure
- Redfshift
- Alibaba Cloud
- Oracle Cloud
- IBM cloud
- ...

Casos de uso, algunos ejemplos

Grandes compañías usan herramientas Big Data en su negocio para diferentes propósitos. Google, Facebook, LinkedIn, Amazon, Inditex, Grupo Santander,... En el libro (Marr, 2016) podemos ver 45 casos de éxito donde se hace uso de herramientas y metodologías Big Data.

- Sector Financiero: detección de fraude (bases de datos graph oriented).
- Sector de hostelería y ocio. Profiling, Sistemas de recomendación.
- Sector de moda. Herramientas de Big Data aplicadas al análisis.
- Redes Sociales.
- Metabuscadores.
- ..

Salidas profesionales

- **Big Data Architect**. Se encarga de deseñar las bases datos y escoger las herramientas más adecuadas al propósito del proyecto. Por lo general se encarga del diseño de la solución a implementar. Debe ser conocedor de las herramientas de manera profunda, conociendo sus puntos fuertes y débiles para escoger lo más apropiado al proyecto.

- **Big Data Engineer**. Se encarga del desarrollo, mantenimiento, pruebas y calificación de las soluciones big data puestas en marcha en el proyecto.

- **Big data Developer**. Se encarga del desarrollo de Software y pruebas unitarias. Debe disponer de altos conocimientos de programación en distintos lenguajes y manejo de frameworks y herramientas Big Data.

- **Data Scientist**. Se encargan de modelar modelos estadísticos y su implementación para explotar los datos y descender información de valor de estos. Debe ser conocedor de los modelos existentes y disponer de gran capacidad de análisis, así como bases matemáticas y de programación en R o python.

- **Data Analyst**. Se encargan sobre todo de la visualización de los datos, realización de informes. Debe ser conocedor de herramientas como Tableu, Qlik, Cognos, SAP y disponer de conocimientos SQL.

- **Chief Data Officer (CDO)**. Responsable de la estrategia Big Data en una organización. Debe ser conocedor del mundo Big Data, así como del negocio y la gestión. Se encarga de supervisar la estrategia Big Data implementada y de su avance.

- **Big Data Consultant**. Asesora a las compañías en estrategias Big Data. Al igual que Data Architect, debe ser conocedor de las herramientas Big Data "por dentro", a demás de conocimientos del negocio para ver los requerimientos y necesidades de la compañía en sus proyectos Big Data. En ocasiones participa en diversas fases tanto en arquitectura, desarrollo y supervisión.

En la próxima sesión

Sesion 02: **Fundamentos de las bases de datos**

- Bases de datos desde dentro. *Data Files* e *Index Files*
- Indices: B+Tree, B+ Cluster, Hash
- Optimización de las consultas
- ...

referencias

- Marr, Bernard (2016). *Big Data in practice.* The Atrium, Southern Gate, Chichester PO19 8SQ, United Kingdom: JohnWiley y Sons Ltd. ISBN: 978-1-119-23138-7.
- Mashey, John R. (1998). *Big Data and the Next Wave of InfraStress.* https://static.usenix.org/event/usenix99/invited_talks/mashey.pdf.
- Strauch, Christof (s.f.). *NoSQL Databases.* https://www.christof-strauch.de/nosqldbs.pdf.