Todo Lo Que un Científico de Datos Debe Saber Sobre la Gestión de Datos (Pero Tiene Miedo de Preguntar)

datasource.ai/es/data-science-articles/todo-lo-que-un-cientifico-de-datos-debe-saber-sobre-la-gestion-de-datos-pero-tiene-miedo-de-preguntar

2 de abril de 2020

Para ser un verdadero científico de datos "full-stack", o lo que muchos blogueros y empleadores llaman un "unicornio", tienes que dominar cada paso del proceso de la ciencia de los datos, desde el almacenamiento de tus datos, hasta la puesta en producción de tu producto final (típicamente un modelo predictivo).

Pero la mayor parte de la formación en data science se centra en técnicas de machine learning o deep learning y el conocimiento en gestión de datos se trata a menudo como una idea a parte.

Los estudiantes de data science suelen aprender habilidades de modelado con datos ya procesados y limpios en archivos de texto almacenados en su portátil, ignorando cómo se hace ese manejo de los datos.

Los estudiantes a menudo no se dan cuenta de que conseguir que los datos (y obtenerlos de varias fuentes) estén listos para el modelado suele ser el 80% del trabajo.

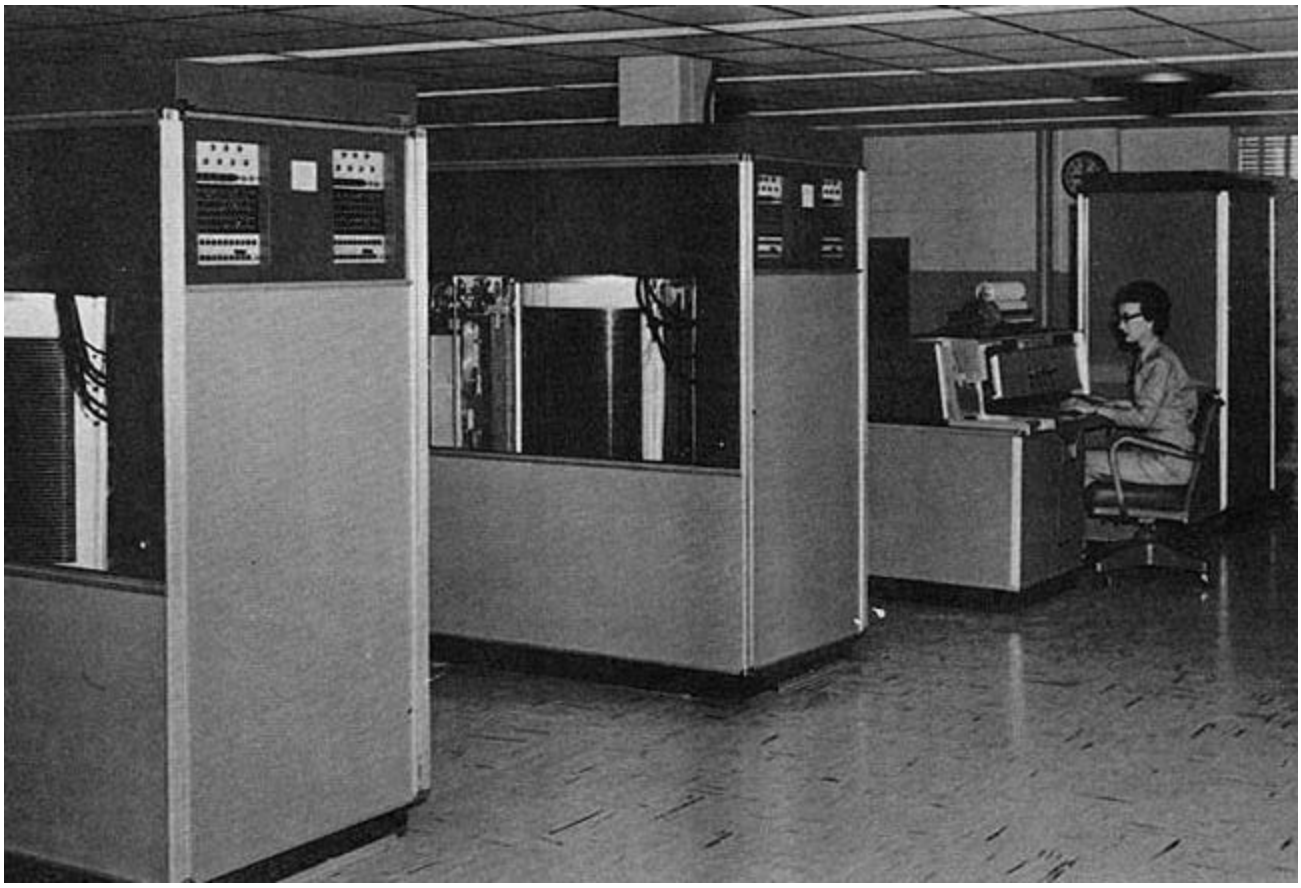
Y debido a que los proyectos de grandes empresas suelen implicar una cantidad masiva de datos, su máquina local no está equipada para manejar todo el proceso de modelado y a menudo se lleva a cabo en la nube, con la mayoría de las aplicaciones y bases de datos alojadas en los servidores de los centros de datos en otros lugares.

Incluso después de que el estudiante consiguió un trabajo como científico de datos, la gestión de datos a menudo se convierte en algo de lo que se encarga un equipo de ingeniería de datos por separado.

Como resultado, demasiados científicos de datos saben muy poco sobre almacenamiento de datos e infraestructura, a menudo en detrimento de su capacidad para tomar las decisiones correctas en sus trabajos.

El objetivo de este artículo es proporcionar una hoja de ruta de lo que un científico de datos en 2020 debe saber sobre la gestión de datos - desde los tipos de bases de datos, dónde y cómo se almacenan y procesan los datos, hasta las opciones comerciales actuales - para que los aspirantes a "unicornios" puedan profundizar por su cuenta, o al menos aprender lo suficiente para parecerlo en entrevistas y eventos de networking.

El aumento de los datos no estructurados y las herramientas de Big Data



IBM 305 RAMAC (Fuente: WikiCommons)

La historia del data science es realmente la historia del almacenamiento de datos. En la era pre-digital, los datos se almacenaban en nuestras cabezas, en tablillas de arcilla o en papel, lo que hacía que la agregación y el análisis de los datos llevara mucho tiempo.

En 1956, IBM introdujo la primera computadora comercial con un disco duro magnético, 305 RAMAC. La unidad completa requería 30 x 50 pies de espacio físico, pesaba más de una tonelada, y por 3.200 dólares al mes, las empresas podían alquilar la unidad para almacenar hasta 5 MB de datos.

En los 60 años transcurridos desde entonces, los precios por gigabyte en DRAM han bajado de la friolera de 2.640 millones de dólares en 1965 a 4.9 en 2017. Además de ser, por mucha diferencia, más baratas, el almacenamiento de datos también se volvió mucho más denso/pequeño en tamaño.

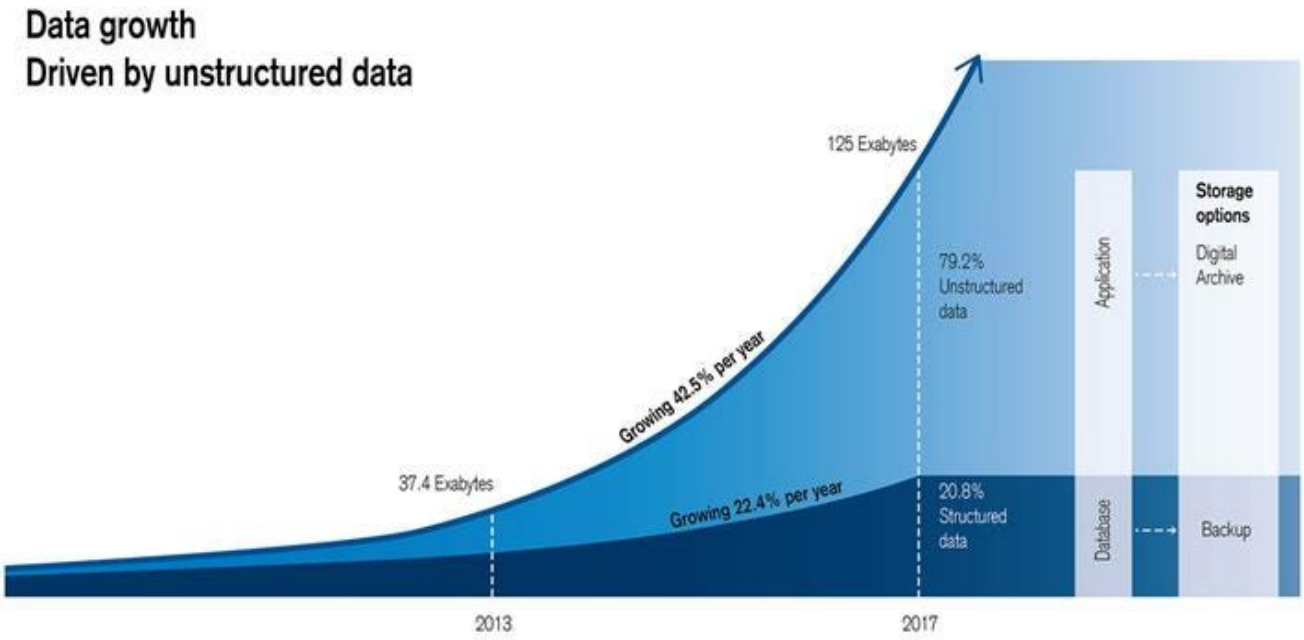
Un plato de disco en la RAMAC 305 almacenaba cien bits por pulgada cuadrada, comparado con más de un trillón de bits por pulgada cuadrada en un plato de disco típico hoy en día.

Esta combinación de costo y tamaño (dramáticamente reducidos) en el almacenamiento de datos es lo que hace posible el gran análisis de datos de hoy en día.

Con un costo de almacenamiento ultra bajo, la construcción de la infraestructura de la ciencia de los datos para recolectar y extraer información de una enorme cantidad de datos se convirtió en un enfoque rentable para las empresas.

Y con la proliferación de los dispositivos IoT que generan y transmiten constantemente los datos de los usuarios, las empresas están recogiendo datos sobre un número cada vez mayor de actividades, creando una cantidad masiva de activos de información de *gran volumen, alta velocidad y gran variedad (o las "tres V del Big Data").*

La mayoría de esas actividades (por ejemplo, correos electrónicos, vídeos, audio, mensajes de chat, publicaciones en medios sociales) generan datos no estructurados, que representan casi el 80% de los datos totales de las empresas en la actualidad y están creciendo el doble de rápido que los datos estructurados en el último decenio.



25 Exabytes de datos empresariales se almacenaron en 2017; el 80% eran datos no estructurados. (Fuente: **Credit Suisse**)

Este crecimiento masivo de datos transformó dramáticamente la forma en que se almacenan y analizan los datos, ya que las herramientas y enfoques tradicionales no estaban equipadas

para manejar las "tres V de los grandes datos". Se desarrollaron nuevas tecnologías con la capacidad de manejar el volumen y la variedad cada vez mayor de datos, y a una velocidad más rápida y a un costo menor.

Estas nuevas herramientas también tienen profundos efectos en la forma en que los científicos de datos hacen su trabajo, permitiéndoles monetizar el masivo volumen de datos mediante la realización de análisis y la construcción de nuevas aplicaciones que antes no eran posibles. A continuación se presentan las principales innovaciones de gestión de datos que creemos que todo científico de datos debe conocer.

Bases de datos relacionales y NoSQL

Los Sistemas de Gestión de Bases de Datos Relacionales (RDBMS - Relational Data Base Management Systems) surgieron en la década de 1970 para almacenar datos en forma de tablas con filas y columnas, utilizando declaraciones del Lenguaje de Consulta Estructurado (SQL) para consultar y mantener la base de datos.

Una base de datos relacional es básicamente una colección de tablas, cada una con un esquema que define rígidamente los atributos y tipos de datos que almacenan, así como claves que identifican columnas o filas específicas para facilitar el acceso.

El panorama de los sistemas de gestión de bases de datos relacionales fue alguna vez gobernado por Oracle e IBM, pero hoy en día muchas opciones de código abierto, como MySQL, SQLite y PostgreSQL son igual de populares.

Rank			DBMS	Database Model	Score		
Aug 2019	Jul 2019	Aug 2018			Aug 2019	Jul 2019	Aug 2018
1.	1.	1.	Oracle	Relational, Multi-model	1339.48	+18.22	+27.45
2.	2.	2.	MySQL	Relational, Multi-model	1253.68	+24.16	+46.87
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1093.18	+2.35	+20.53
4.	4.	4.	PostgreSQL	Relational, Multi-model	481.33	-1.94	+63.83
5.	5.	5.	IBM Db2	Relational, Multi-model	172.95	-1.19	-8.89
6.	6.	6.	Microsoft Access	Relational	135.33	-1.98	+6.24
7.	7.	7.	SQLite	Relational	122.72	-1.91	+8.99
8.	8.	↑ 9.	MariaDB	Relational, Multi-model	84.95	+0.52	+16.66
9.	9.	↑ 11.	Hive	Relational	81.80	+0.93	+23.86
10.	10.	↓ 8.	Teradata	Relational, Multi-model	76.64	-1.18	-0.77
11.	11.	↑ 12.	FileMaker	Relational	58.02	+0.12	+1.96
12.	12.	↓ 10.	SAP Adaptive Server	Relational	55.86	-0.79	-4.57
13.	13.	13.	SAP HANA	Relational, Multi-model	55.43	-0.11	+3.50
14.	14.	14.	Microsoft Azure SQL Database	Relational, Multi-model	27.99	-0.67	+1.89
15.	15.	15.	Informix	Relational, Multi-model	25.68	-0.18	+0.29
16.	16.	↑ 20.	Google BigQuery	Relational	24.47	+0.55	+10.06
17.	17.	17.	Vertica	Relational, Multi-model	23.80	+0.96	+3.76
18.	↑ 19.	↑ 19.	Amazon Redshift	Relational	22.61	+1.68	+7.43
19.	↑ 20.	↓ 18.	Netezza	Relational	20.84	+0.23	+4.50
20.	↓ 18.	↓ 16.	Firebird	Relational	20.51	-0.88	+0.22
21.	↑ 22.	↑ 24.	dBASE	Relational	16.88	+0.24	+6.75
22.	↓ 21.	22.	Spark SQL	Relational	16.34	-0.41	+3.54
23.	23.	↓ 21.	Impala	Relational, Multi-model	15.07	+0.21	+1.57
24.	24.	↓ 23.	Greenplum	Relational, Multi-model	12.77	+0.28	+2.39
25.	25.	25.	Oracle Essbase	Relational	12.41	+0.71	+4.34

RDBMS rankeados por popularidad (Fuente: **DB-Engines**)

Las bases de datos relacionales encontraron un hogar en el mundo de los negocios debido a algunas propiedades muy atractivas. La integridad de los datos es absolutamente primordial en las bases de datos relacionales.

Los RDBMS satisfacen los requisitos de atomicidad, consistencia, aislamiento y durabilidad (o conformes con el ACID) imponiendo una serie de restricciones para asegurar que los datos almacenados sean fiables y precisos, lo que los hace ideales para el seguimiento y almacenamiento de cosas como números de cuenta, pedidos y pagos. Pero estas restricciones vienen con costosas compensaciones.

Debido a las restricciones de esquema y tipo, los RDBMS son terribles para almacenar datos no estructurados o semiestructurados.

El rígido esquema también hace que los RDBMS sean más caros de configurar, mantener y crecer. La configuración de un RDBMS requiere que los usuarios dispongan de casos de uso específicos con antelación; cualquier cambio en el esquema suele ser difícil y lleva mucho

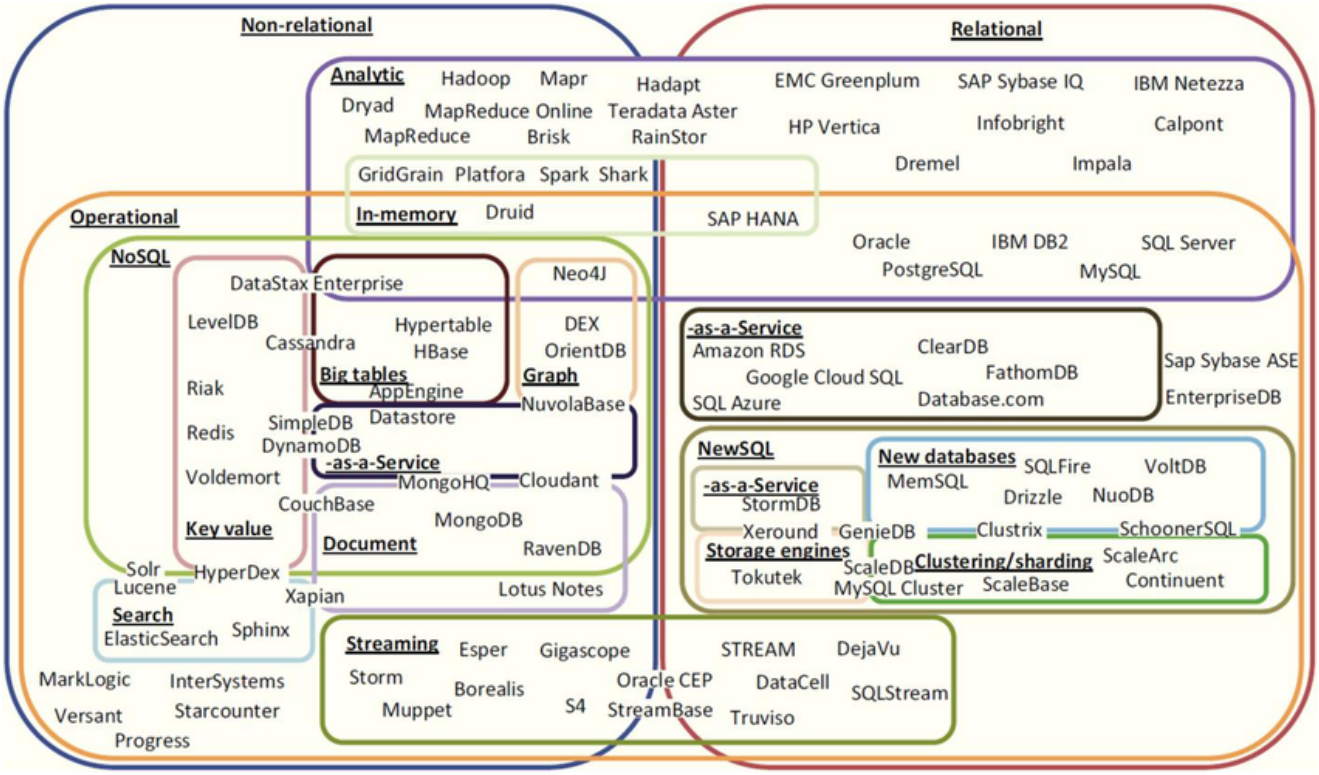
tiempo.

Además, los RDBMS tradicionales se diseñaron para funcionar en un solo nodo de computadora, lo que significa que su velocidad es significativamente más lenta cuando se procesan grandes volúmenes de datos. También es sumamente difícil cambiar el RDBMS para escalar horizontalmente y mantener al mismo tiempo el cumplimiento del ACID. Todos estos atributos hacen que los RDBMS tradicionales estén mal equipados para manejar el big data moderno.

A mediados de los años 2000, los RDBMS existentes ya no podían manejar las necesidades cambiantes y el crecimiento exponencial de unos pocos negocios en línea muy exitosos, y muchas bases de datos no relacionales (o NoSQL) se desarrollaron como resultado (aquí hay una historia sobre cómo Facebook se enfrentó a las limitaciones de MySQL cuando su volumen de datos comenzó a crecer).

Sin ninguna solución conocida en ese momento, estos negocios en línea inventaron nuevos enfoques y herramientas para manejar la enorme cantidad de datos no estructurados que recolectaban: Google creó GFS, MapReduce y BigTable; Amazon creó DynamoDB; Yahoo creó Hadoop; Facebook creó Cassandra y Hive; LinkedIn creó Kafka.

Algunas de estas empresas lanzaron éste trabao en código abierto (open source); algunas publicaron documentos de investigación en los que detallaban sus diseños, lo que dio lugar a una proliferación de bases de datos con las nuevas tecnologías, y las bases de datos NoSQL se convirtieron en un importante protagonista de la industria.



Una explosión de opciones de bases de datos desde los años 2000. Fuente: **Korflatis et. al (2016)**

Las bases de datos NoSQL son agnósticas al esquema y proporcionan la flexibilidad necesaria para almacenar y manipular grandes volúmenes de datos no estructurados y semiestructurados.

Los usuarios no necesitan saber qué tipos de datos se almacenarán durante la configuración, y el sistema puede acomodar los cambios en los tipos de datos y el esquema.

Diseñadas para distribuir los datos a través de diferentes nodos, las bases de datos NoSQL son generalmente más escalables horizontalmente y tolerantes a las fallas.

Sin embargo, estos beneficios de rendimiento también tienen un costo: las bases de datos NoSQL no son compatibles con el ACID y la consistencia de los datos no está garantizada. En cambio, proporcionan una "consistencia eventual": cuando los datos antiguos se sobrescriben, devuelven resultados que están un poco equivocados temporalmente.

Por ejemplo, el índice del motor de búsqueda de Google no puede sobrescribir sus datos mientras la gente está buscando simultáneamente un término dado, por lo que no nos da los resultados más actualizados cuando buscamos, pero nos da la última y mejor respuesta posible.

Aunque esta configuración no funcionará en situaciones en las que la consistencia de los datos sea absolutamente necesaria (como en las transacciones financieras), está bien para las tareas que requieren rapidez en lugar de precisión.

Ahora hay varias categorías diferentes de NoSQL, cada una sirviendo a algunos propósitos específicos. Los Key-Value Stores, como Redis, DynamoDB y Cosmos DB, almacenan sólo pares clave-valor y proporcionan una funcionalidad básica para recuperar el valor asociado a una clave conocida.

Funcionan mejor con un esquema simple de base de datos y cuando la velocidad es importante. Las Wide Column Stores, como **Cassandra**, **Scylla**, y **HBase**, almacenan datos en familias de columnas o tablas, y están construidas para manejar petabytes de datos a través de un sistema masivo y distribuido.

Document Stores, como **MongoDB** y **Couchbase**, almacenan datos en formato XML o JSON, con el nombre del documento como clave y el contenido del documento como valor.

Los documentos pueden contener muchos tipos de valores diferentes y pueden anidarse, lo que los hace especialmente adecuados para gestionar datos semiestructurados en sistemas distribuidos.

Las bases de datos de grafos, como **Neo4J** y **Amazon Neptune**, representan los datos como una red de nodos u objetos relacionados para facilitar la visualización de los datos y el análisis de los gráficos.

Las bases de datos de grafos son particularmente útiles para analizar las relaciones entre puntos de datos heterogéneos, como en la prevención de fraudes o el gráfico de amigos de Facebook (Facebook Graph).

MongoDB es actualmente la base de datos NoSQL más popular, y ha proporcionado un valor agregado sustancial para algunas empresas que han estado luchando por manejar sus datos no estructurados con el enfoque tradicional de RDBMS.

He aquí dos ejemplos de la industria: después de que MetLife pasó años tratando de construir una base de datos centralizada de clientes en un RDBMS que pudiera manejar todos sus productos de seguros, alguien en una hackathon interna construyó una con MongoDB en cuestión de horas, y entró en producción en 90 días.

YouGov, una empresa de investigación de mercado que recoge 5 gigabits de datos por hora, ahorró el 70 por ciento de la capacidad de almacenamiento que antes utilizaba al migrar de RDBMS a MongoDB.

Data Warehouse, Data Lake, y Data Swamp

A medida que las fuentes de datos siguen creciendo, la realización de análisis de datos con múltiples bases de datos se volvió ineficiente y costosa. En los años 80 surgió una solución llamada **Data Warehouse**, que centraliza los datos de una empresa de todas sus bases de datos.

Data Warehouse soporta el flujo de datos desde los sistemas operativos a los sistemas de análisis/decisión creando un único repositorio de datos de varias fuentes (tanto internas como externas). En la mayoría de los casos, un Data Warehouse es una base de datos relacional que almacena los datos procesados y que está optimizada para recopilar información empresarial.

Recoge datos con una estructura y un esquema predeterminados procedentes de sistemas transaccionales y aplicaciones empresariales, y los datos se utilizan normalmente para la presentación de informes y análisis operacionales.

Pero debido a que los datos que entran en los almacenes de datos necesitan ser procesados antes de ser almacenados - con la enorme cantidad de datos no estructurados de hoy en día, eso podría llevar mucho tiempo y recursos.

En respuesta, las empresas comenzaron a mantener **Data Lakes** en la década de 2010, que almacenan todos los datos estructurados y no estructurados de una empresa a cualquier escala. Los Data Lakes almacenan datos en bruto, y se pueden configurar sin tener que

definir primero la estructura y el esquema de datos.

Los Data Lakes permiten a los usuarios ejecutar análisis sin tener que trasladar los datos a un sistema de análisis separado, lo que permite a las empresas obtener conocimientos de nuevas fuentes de datos que antes no estaban disponibles para el análisis, por ejemplo, construyendo modelos de machine learning utilizando datos de archivos de registro, flujos de clics, medios sociales y dispositivos IoT.

Al hacer que todos los datos de la empresa estén disponibles para el análisis, los científicos de datos podrían responder a un nuevo conjunto de preguntas de negocios, o abordar las viejas preguntas con nuevos datos.

Characteristics	Data Warehouse	Data Lake
Data	Relational from transactional systems, operational databases, and line of business applications	Non-relational and relational from IoT devices, web sites, mobile apps, social media, and corporate applications
Schema	Designed prior to the DW implementation (schema-on-write)	Written at the time of analysis (schema-on-read)
Price/Performance	Fastest query results using higher cost storage	Query results getting faster using low-cost storage
Data Quality	Highly curated data that serves as the central version of the truth	Any data that may or may not be curated (ie. raw data)
Users	Business analysts	Data scientists, Data developers, and Business analysts (using curated data)
Analytics	Batch reporting, BI and visualizations	Machine Learning, Predictive analytics, data discovery and profiling

Comparación entre Data Warehouse y Data Lake (Fuente: **AWS**)

Un desafío común con la arquitectura de los Data Lakes es que sin la calidad de datos apropiada y el marco de gobierno (governance framework) en su lugar, cuando los terabytes de datos estructurados y no estructurados fluyen hacia los Data Lakes a menudo se hace extremadamente difícil clasificar su contenido.

Los Data Lakes podrían convertirse en Data Swamps, ya que los datos almacenados se vuelven demasiado desordenados para ser utilizables. Muchas organizaciones están pidiendo ahora más prácticas de gobierno de datos y de gestión de metadatos para evitar que se formen los Data Swamps.

Procesamiento distribuido y paralelo: Hadoop, Spark y MPP

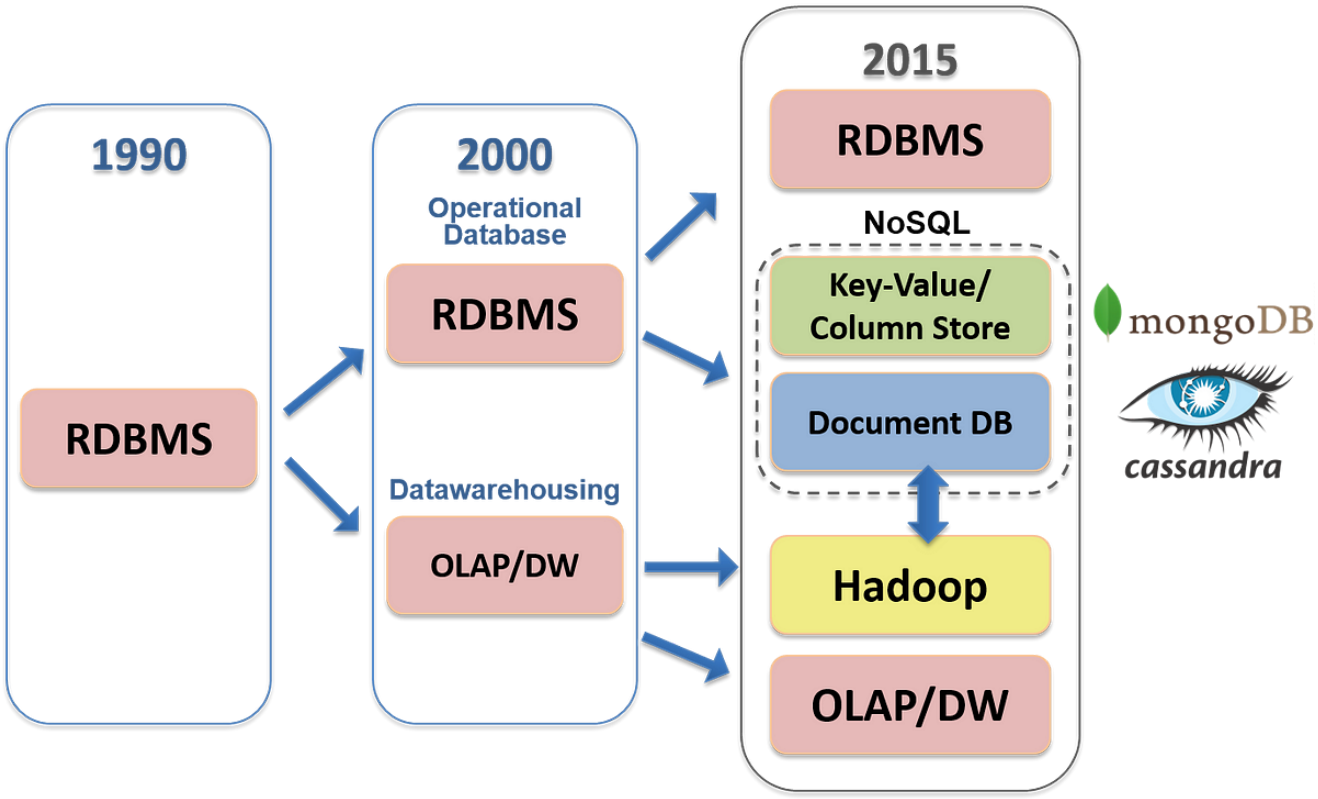
Mientras que las necesidades de almacenamiento y computación crecieron a pasos agigantados en las últimas décadas, el hardware tradicional no ha avanzado lo suficiente para mantenerse al día.

Los datos de la empresa ya no encajan perfectamente en el almacenamiento estándar, y la potencia de cálculo necesaria para manejar la mayoría de las grandes tareas de análisis de datos puede llevar semanas, meses, o simplemente no es posible completarlas en un ordenador estándar.

Para superar esta deficiencia, muchas nuevas tecnologías han evolucionado para incluir múltiples computadoras que trabajan juntas, distribuyendo la base de datos a miles de servidores básicos/sencillos. Cuando una red de computadoras se conectan y trabajan juntas para realizar la misma tarea, las computadoras forman un clúster.

Se puede pensar en un clúster como una sola computadora, pero puede mejorar drásticamente el rendimiento, la disponibilidad y la escalabilidad en una sola máquina más potente y a un costo menor mediante el uso de hardware básico.

Apache Hadoop es un ejemplo de infraestructuras de datos distribuidos que aprovechan los clusters para almacenar y procesar cantidades masivas de datos, y lo que permite la arquitectura Data Lake.



Evolucion de las tecnologías en bases de datos (Source: **Business Analytic 3.0**)

Cuando pienses en Hadoop, piensa en "distribución". Hadoop consta de tres componentes principales: Hadoop Distributed File System (HDFS), una forma de almacenar y hacer un seguimiento de sus datos a través de múltiples discos duros físicos (distribuidos); MapReduce, un marco para procesar datos a través de procesadores distribuidos; y Yet Another Resource Negotiator (YARN), un marco de gestión de clústeres que orquesta la distribución de cosas como el uso de la CPU, la memoria y la asignación de ancho de banda de red a través de ordenadores distribuidos.

La capa de procesamiento de Hadoop es una innovación especialmente notable: MapReduce es un enfoque computacional de dos pasos para procesar grandes conjuntos de datos (de varios terabytes o más) distribuidos a través de grandes clusters de hardware básico de una manera fiable y tolerante a las fallas.

El primer paso es distribuir los datos en múltiples computadoras (Map), y cada una de ellas realiza un cálculo en su parte de los datos en paralelo.

El siguiente paso es combinar esos resultados de una manera "en pareja - pair-wise" (Reducir). Google publicó un paper sobre MapReduce en 2004, que fue recogido por los programadores de Yahoo que lo implementaron en el entorno Apache de código abierto en 2006, proporcionando a todas las empresas la capacidad de almacenar un volumen de datos sin precedentes utilizando hardware de consumo.

Aunque hay muchas implementaciones de código abierto sobre ésta misma idea, la marca MapReduce de Google se ha mantenido, algo así como las marcas Jacuzzi o Kleenex.

Hadoop está construido para cálculos iterativos, escaneando cantidades masivas de datos en una sola operación desde el disco, distribuyendo el procesamiento a través de múltiples nodos, y almacenando los resultados de nuevo en el disco.

La consulta de zettabytes de datos indexados que tomaría 4 horas para ejecutarse en un entorno de almacenamiento de datos tradicional podría completarse en 10-12 segundos con Hadoop y HBase. Hadoop se utiliza típicamente para generar modelos analíticos complejos o aplicaciones de almacenamiento de datos de gran volumen, como análisis retrospectivos y predictivos; machine learning y correspondencia de patrones (pattern matching); segmentación de clientes y análisis de rotación.

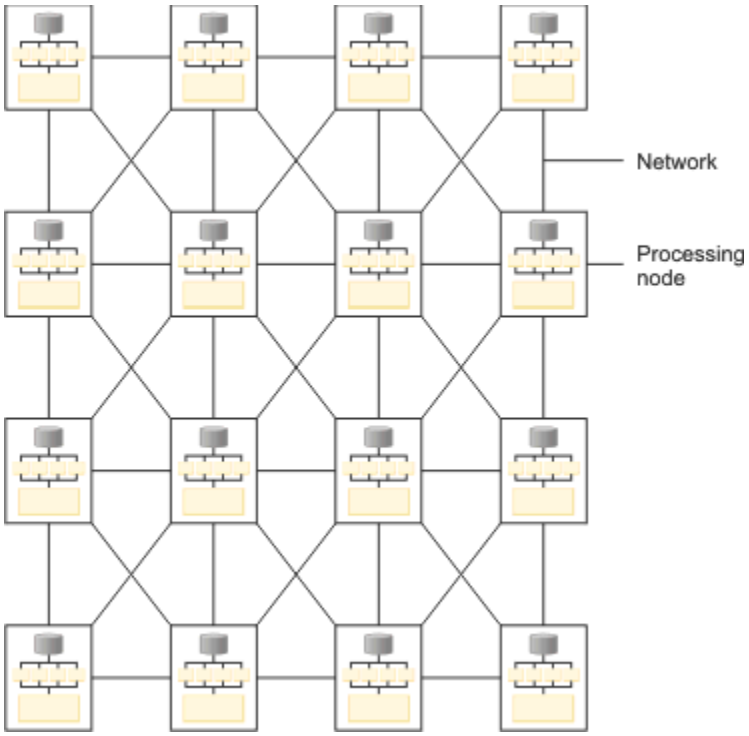
Pero MapReduce procesa los datos en lotes y, por lo tanto, no es adecuado para procesar datos en tiempo real. Apache Spark fue construido en 2012 para llenar ese vacío.

Spark es una herramienta de procesamiento de datos en paralelo que se optimiza para obtener velocidad y eficiencia mediante el procesamiento de datos en memoria. Funciona bajo el mismo principio de MapReduce, pero se ejecuta mucho más rápido completando la mayor parte del cómputo en memoria y sólo escribiendo en disco cuando la memoria está llena o el cómputo se ha completado.

Este cómputo en memoria permite a Spark "ejecutar programas hasta 100 veces más rápido que Hadoop MapReduce en memoria, o 10 veces más rápido en disco". Sin embargo, cuando el conjunto de datos es tan grande que la insuficiencia de la memoria RAM se convierte en un problema (por lo general, cientos de gigabytes o más), Hadoop MapReduce podría superar a Spark.

Spark también tiene un extenso conjunto de bibliotecas de análisis de datos que cubren una amplia gama de funciones: Spark SQL para SQL y datos estructurados; MLib para machine learning, Spark Streaming para el procesamiento de stream y GraphX para el análisis de gráficos.

Como Spark se centra en la computación, no viene con su propio sistema de almacenamiento y en su lugar se ejecuta en una variedad de sistemas de almacenamiento como Amazon S3, Azure Storage, y HDFS de Hadoop.



En un sistema MPP, todos los nodos están interconectados y los datos pueden ser intercambiados a través de la red (Fuente: **IBM**)

Hadoop y Spark no son las únicas tecnologías que aprovechan los clusters para procesar grandes volúmenes de datos.

Otro enfoque computacional popular para el procesamiento distribuido de consultas se llama Procesamiento Masivo en Paralelo (MPP).

De manera similar a MapReduce, MPP distribuye el procesamiento de datos entre múltiples nodos, y los nodos procesan los datos en paralelo para una mayor velocidad. Pero a diferencia de Hadoop, MPP se utiliza en el RDBMS y utiliza una arquitectura de "no compartir nada - share nothing" - cada nodo procesa su propia porción de los datos con procesadores de múltiples núcleos, lo que los hace mucho más rápidos que el RDBMS tradicional.

Algunas bases de datos MPP, como Pivotal Greenplum, tienen librerías de machine learning maduras que permiten el análisis dentro de la base de datos. Sin embargo, al igual que con el RDBMS tradicional, la mayoría de las bases de datos MPP no admiten datos no estructurados, e incluso los datos estructurados requerirán algún tipo de procesamiento para adaptarse a la infraestructura MPP; por lo tanto, se necesita tiempo y recursos adicionales para establecer la tubería de datos (pipeline) para una base de datos MPP.

Dado que las bases de datos MPP cumplen con los requisitos del ACID y ofrecen una velocidad mucho más rápida que los RDBMS tradicionales, suelen emplearse en soluciones de almacenamiento de datos empresariales de alto nivel, como Amazon Redshift, Pivotal Greenplum y Snowflake. Como ejemplo de la industria, la Bolsa de Valores de Nueva York recibe de cuatro a cinco terabytes de datos diariamente y lleva a cabo complejos análisis, vigilancia del mercado, planificación de la capacidad y supervisión.

La empresa había estado utilizando una base de datos tradicional que no podía manejar la carga de trabajo, que se cargaba en horas y tenía poca velocidad de consulta. El cambio a una base de datos MPP redujo el tiempo de ejecución de sus análisis diarios en ocho horas.

Servicios en la nube

Otra innovación que transformó completamente las capacidades de análisis de datos de las grandes empresas es el auge de los servicios en la nube.

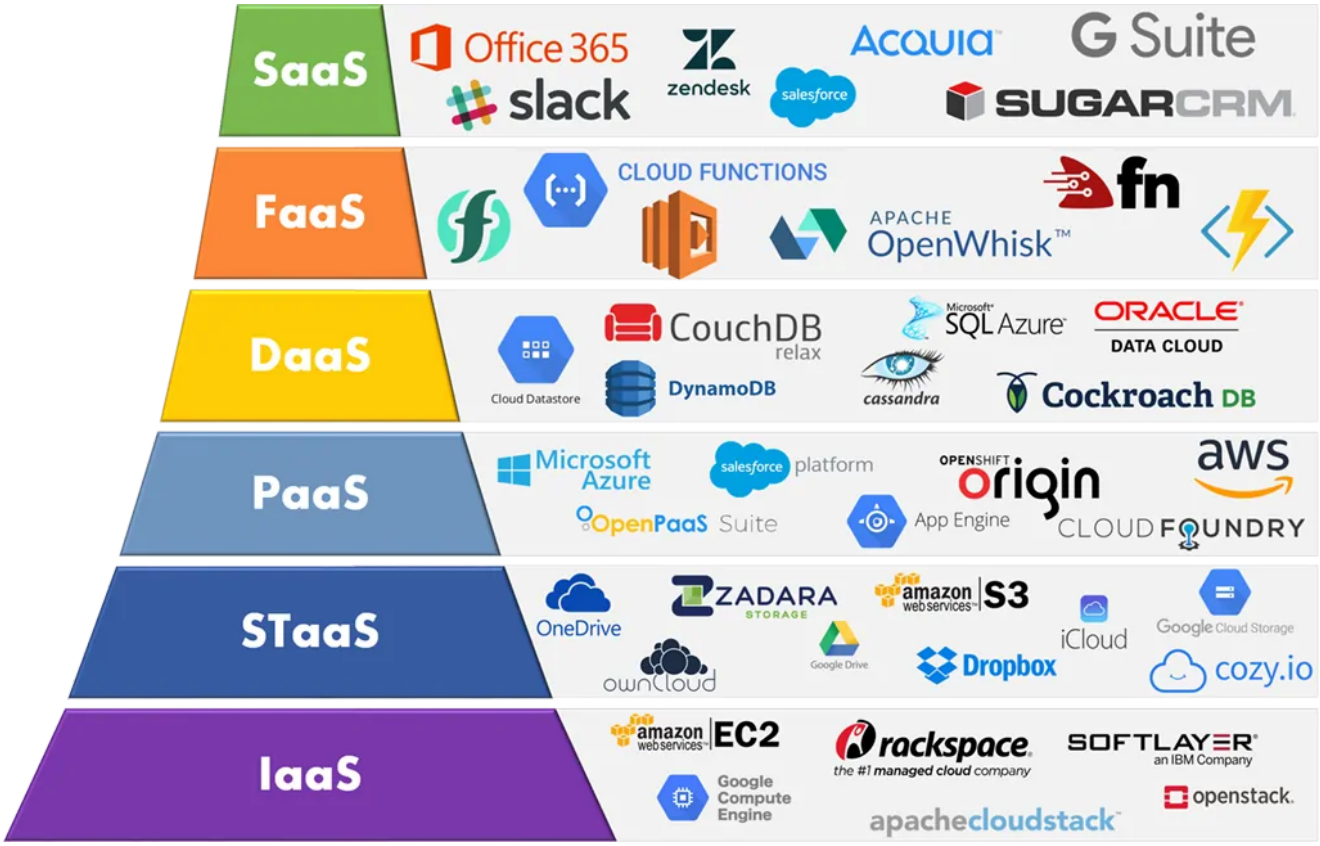
En los malos tiempos antes de que los servicios de nube estuvieran disponibles, las empresas tenían que comprar soluciones de almacenamiento y análisis de datos en las instalaciones de los proveedores de software y hardware, normalmente pagando por adelantado las tasas de licencia perpetua del software y las tasas anuales de mantenimiento y servicio de hardware.

A esto se suman los costos de energía, refrigeración, seguridad, protección contra desastres, personal de TI, etc., para la construcción y mantenimiento de la infraestructura en las instalaciones.

Incluso cuando era técnicamente posible almacenar y procesar grandes datos, la mayoría de las empresas encontraban prohibitivo el costo de hacerlo a escala.

La ampliación de la infraestructura en las instalaciones también requiere un extenso proceso de diseño y adquisición, cuya aplicación lleva mucho tiempo y requiere un capital inicial

considerable. Como resultado, se ignoraron muchas posibilidades de recopilación y análisis de datos potencialmente valiosos.



Proveedores de "As a Service": por ejemplo, Infraestructura como un Servicio (IaaS) y Almacenamiento como un Servicio (STaaS) (Fuente: **IMELGRAT.ME**)

El modelo de crear infraestructura en las propias instalaciones (on-promises) comenzó a perder cuota de mercado rápidamente cuando se introdujeron los servicios en la nube a finales del 2000 - el mercado mundial de servicios en la nube ha estado creciendo un 15% anual en la última década.

Las plataformas de servicios en la nube ofrecen suscripciones a una variedad de servicios (desde computación virtual hasta infraestructura de almacenamiento y bases de datos), que se ofrecen a través de Internet sobre una base de pago por uso, ofreciendo a los clientes un acceso rápido a recursos de almacenamiento y computación virtual flexibles y de bajo costo.

Los proveedores de servicios en la nube son responsables de todas sus compras y mantenimiento de hardware y software, y suelen disponer de una amplia red de servidores y personal de apoyo para prestar servicios fiables.

Muchas empresas descubrieron que podían reducir significativamente los costos y mejorar la eficiencia operacional con los servicios en la nube, y son capaces de desarrollar y producir sus productos más rápidamente con los recursos de la nube listos para usar y su escalabilidad incorporada.

Al eliminar los costos iniciales y el compromiso de tiempo para construir la infraestructura en sus propias instalaciones, los servicios de nube también disminuyen las barreras para adoptar herramientas de big data, y democratizaron efectivamente el análisis big data para las pequeñas y medianas empresas.

Existen varios modelos de servicios en la nube, siendo las nubes públicas las más comunes. En una nube pública, todo el hardware, software y otra infraestructura de apoyo es propiedad del proveedor de servicios en la nube y es administrada por él.

Los clientes comparten la infraestructura en la nube con otros "inquilinos de la nube" y acceden a sus servicios a través de un navegador de Internet.

Una nube privada suele ser utilizada por organizaciones con necesidades especiales de seguridad, como organismos gubernamentales e instituciones financieras. En una nube privada, los servicios y la infraestructura se dedican exclusivamente a una organización y se mantienen en una red privada.

La nube privada puede ser local o alojada por un tercer proveedor de servicios en otro lugar. Las nubes híbridas combinan las nubes privadas con las nubes públicas, lo que permite a las organizaciones aprovechar las ventajas de ambas.

En una nube híbrida, los datos y las aplicaciones pueden desplazarse entre las nubes privadas y públicas para lograr una mayor flexibilidad: por ejemplo, la nube pública podría utilizarse para datos de gran volumen y menor seguridad, y la nube privada para datos sensibles y críticos para el negocio, como los informes financieros.

El modelo de nubes múltiples implica múltiples plataformas de nubes, cada una de las cuales ofrece un servicio de aplicación específico. Una multi-nube puede ser una combinación de nubes públicas, privadas e híbridas para lograr los objetivos de la organización.

Las organizaciones a menudo eligen la multi-nube para adaptarse a sus negocios, ubicaciones y necesidades de tiempo particulares, y para evitar el bloqueo del proveedor.

Si quieres conocer un caso de estudio acerca del data management, ve al siguiente link.

Por Phoebe Wong y Robert Bennett

> “Todo Lo Que un Científico de Datos Debe Saber Sobre la Gestión de Datos (Pero Tiene Miedo de Preguntar)”
>
> – Phoebe Wong

Créditos

Este contenido fue escrito originalmente en el siguiente artículo Todo Lo Que un Científico de Datos Debe Saber Sobre la Gestión de Datos (Pero Tiene Miedo de Preguntar) y fue traducido al español y de forma literal por el equipo de DataSource.ai con el permiso de su autor original Phoebe Wong y puede encontrar el perfil de esta persona en el siguiente link Phoebe Wong . Este post fue traducido por: Daniel Morales

Data Science

Jan 06, 2023

Top 2 Online Data Science Courses to Improve your Career in 2023

The discipline of Data Science is expanding quickly and has enormous promise. It is used in various sectors, including manufacturing, retail, healt...

Por nikolaos_rbkd-es

Deep learning

Nov 12, 2021

¿Cuándo Es Mejor Evitar el Uso de Deep Learning?

IntroducciónEste artículo está dirigido a los científicos de datos que pueden considerar el uso de algoritmos de aprendizaje profundo, y quieren sa...

Por Matt Przybyla

Data Science

Nov 25, 2021

5 Consejos para Superar Una Entrevista de Trabajo para una Vacante de Científico de Datos

5 Tips To Ace Your Job Interview For A Data Scientist Opening.PNG 795.94 KBImage SourceLos aspirantes a científicos de datos tienen un futuro brill...

Por Daniel Morales