Introducción a Big Data.

Caso práctico

Kindel Media (Dominio público)

FL Logistics, la empresa que **Felisa** y **Luís** crearon hace ya 15 años, no para de crecer.

Acaban de abrir su décimo almacen en España, siguen contratando nuevo personal y cada vez reciben más trabajo.

Sin embargo, precísamente ese crecimiento que para ellos siempre fue un sueño a alcanzar se está convirtiendo en una gran preocupación. Aunque el cometido de su compañía pueda verse como el mover mercancías de un lugar de origen a otro de destino, el funcionamiento de la misma depende en gran medida de ser capaces de capturar, almacenar, gestionar y a ser posible también analizar una gran cantidad de datos (de proveedores, productos, orígenes, destinos, rutas, empleados, ...).

Para el primer almacén que abrieron compraron un servidor de los más potentes del mercado, con el cuál dieron también soporte en un principio al segundo almacén.

Al abrir el tercer almacén vieron que el servidor no era capaz de trabajar con tal carga de información, por lo que optaron por poner un servidor distinto en cada almacén. Eso complicó las cosas porque además necesitaron otro servidor extra en el que poder centralizar trabajo y los resultados de los análisis realizados por los servidores de los diversos almacenes.

Ahora que están abriendo el décimo almacén, el servidor central ya está saturado, lo cual parece imponer un límite al crecimiento de la empresa.

¿Cómo van a poder tratar con tal cantidad de información si las cosas siguen yendo bien y necesitan seguir abriendo más almacenes?

En esta unidad de trabajo vamos a comenzar a conocer el fenómeno conocido como Big Data (o macrodatos).

Comenzaremos entendiendo las circunstancias que originan la aparición de las metodologías y tecnologías para Big Data y veremos qué conseguimos gracias a su uso.

A continuación veremos qué es un clúster de computadoras, los cuales constituyen la infraestructura básica en la que se apoyan los sistemas Big Data.

Más adelante repasaremos una serie de conceptos muy importantes a la hora de terminar de comprender todo el fenómeno Big Data, tanto relacionados con almacenamiento de datos como con procesamiento de los mismos.

Por útimo veremos cuál es la arquitectura en capas que generalmente se utiliza en proyectos Big Data para y terminaremos viendo lo que viene en llamarse el paisaje de Big Data.

1.- Por qué Big Data.

Caso práctico

Roberto y **Luisa** están como casi todos los días de diario comiendo en la oficina.

geralt (Dominio público)

Las cosas en la empresa ya no son como eran. Antes se dedicaban a mover papeles (casi literalmente) y ahora el contenido de todos esos papeles se encuentra en los discos duros de los diversos ordenadores con los que cuenta la empresa.

Pero eso no es todo. Ahora también utilizan grandes cantidades de información que la empresa consigue de fuentes libres o que compra a otras empresas.

Por eso mañana a primera hora se incorpora **Juan** al equipo. **Juan** es experto en Big Data.

—No acabo de entender muy bien cuál es la necesidad de todo esto —dice Roberto.

—Hay una necesidad muy clara. Déjame que te cuente ... —dice Luisa.

En resumen, las metodologías y tecnologías para Big Data aparecen como respuesta la necesidad de tratar cantidades de datos tan grandes que desbordan los sistemas convencionales monomáquina.

Debes conocer

Debes saber que, para complementar la información que veremos en este curso sobre Big Data, existen multitud de fuentes online.

La primera (y muy útil), referencia sería el artículo de la Wikipedia sobre lo que son los macrodatos (traducción al castellano de Big Data).

Macrodatos

Es igualmente muy importante que sepas que la literatura sobre Big Data (al igual que ocurre con las ciencias de la computación en general), es mas

abundante en inglés que en castellano. Por ello, si quieres profundizar por completo por lo general tendrás que acceder a la versión en inglés de la información.

Un ejemplo es el enlace equivalente al previamente indicado pero en la versión inglesa de la Wikipedia el cual contiene más información aún (de hecho si vas a optar por leer sólo uno de ellos, siempre opta por el artículo en inglés).

Big data (en inglés)

Autoevaluación

¿Que problema de base origina la aparición de las metodologías y tecnologías Big Data?

- El tener datos que no se sabe de dónde proceden.
- El tener grandes cantidades de datos que no caben en el almacenamiento conjunto de varias máquinas.
- El tener grandes cantidades de datos que desbordan los recursos de máquinas individuales.
- La incapacidad de realizar analítica en una única máquina.

Incorrecto. Por lo general se sabe bien de dónde proceden.

Incorrecto. Usando clústers de máquinas podemos tratar con ellos.

Correcto.

Incorrecto. Big Data no es sólo para analítica.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

1.1.- Las 5 Vs.

En esta sección hablaremos de las cinco características de Big Data que suelen emplearse para discernir si el procesamiento de datos que necesitamos realizar puede realmente considerarse como Big Data.

José Manuel Fernández Díaz (Dominio público)

Descripción de la imagen

A lo largo de la literatura existente acerca de Big Data, al principio se usaba el término "las 3 Vs", que eran:

- **Volumen.**
- **Velocidad.**
- **Variedad.**

Con el tiempo, a esas 3 características se le han añadido otras 2 características, con lo que han venido en llamarse "las 5 Vs".

- **Volumen.**
- **Velocidad.**
- **Variedad.**
- **Veracidad.**
- **Valor.**

En los últimos años, se han añadido otras 2 características, con lo que en la actualidad cada vez se usa más el término "las 7 Vs":

- **Volumen.**
- **Velocidad.**
- **Variedad.**
- **Veracidad.**
- **Valor.**
- **Viabilidad.** Se trata de la capacidad que tienen las compañías en generar un uso eficaz del gran volumen de datos que manejan.
- **Visualización.** Ningún volumen de datos tan grande

geralt (Dominio público)

tendría sentido si no es posible visualizarlo de manera adecuada. Hacer comprensible y sencilla la lectura de esta información es vital para poder sacar el máximo provecho.

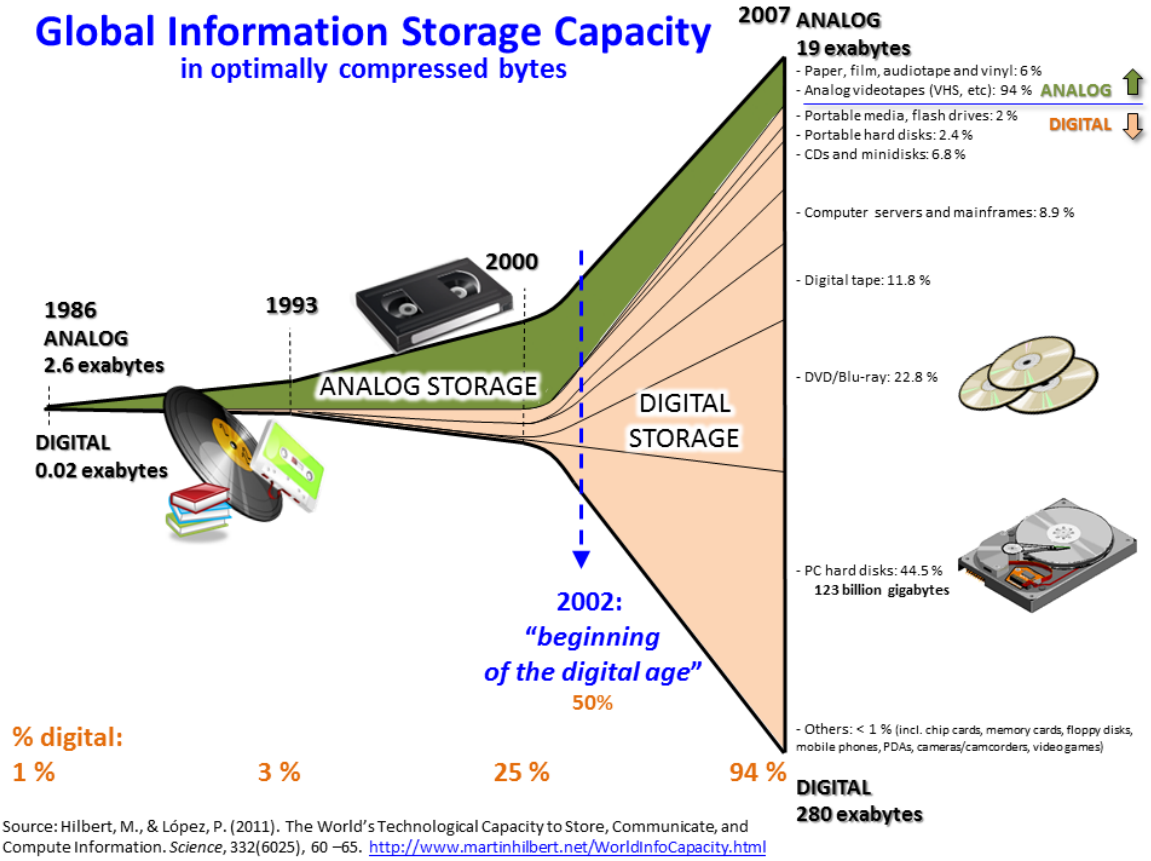
Nosotros vamos a centrarnos en este curso en las **5 Vs**, ya que es la referencia más extendida y consensuada en la literatura.

1.1.1.- Volumen.

La primera característica del reto de tratamiento de datos que ha venido en llamarse Big Data es el volumen de los mismos, es decir, la gran cantidad de bytes de información que los componen.

En la siguiente imagen podemos ver cómo ha ido creciendo en los últimos tiempos la cantidad de información almacenada por el ser humano, gracias a lo cual podemos hacernos una idea de la magnitud del reto.

Capacidad de información global



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

Llegados a este punto, y para poder hacernos una idea de cuál es la magnitud de las cantidades de información con las que es necesario tratar, debemos hacer referencia al significado de las nomenclaturas que se emplean a tal efecto.

Unidades de cantidad de información digital

Unidades	Significado
Bit	Unidad mínima de información en un sistema de computación (almacena un "0" o un "1").
byte (B)	8 bits

kilobyte (kB)	1000 bytes (10^3 bytes)
megabyte (MB)	1000 kilobytes (10^6 bytes)
gigabyte (GB)	1000 megabytes (10^9 bytes)
terabyte (TB)	1000 gigabytes (10^{12} bytes)
petabyte (PB)	1000 terabytes (10^{15} bytes)
exabyte (EB)	1000 petabytes (10^{18} bytes)
zettabyte (ZB)	1000 exabytes (10^{21} bytes)
yottabyte (YB)	1000 zettabytes (10^{24} bytes)

Hay que tener en cuenta que si bien el significado de un kilobyte (kB) es 1000 bytes, dado que en ambientes de computacionales se emplea constantemente la numeración en base 2 también existe el kibibyte (KiB), el cual corresponde a 1024 bytes (2^{10}). De igual modo, también existe el mebibyte (MiB = 2^{20} bytes), el gibibyte (GiB = 2^{30} bytes), y así toda la progresión hasta llegar al yobibyte (YiB = 2^{80} bytes).

Lo que puede producir algo de confusión es que por lo general se emplean las nomenclaturas en base 10 de forma indistinta para designar tanto a la de base 10 como a la (más o menos equivalente) de base 2. Es decir, cuando vemos 1 MB, es posible que signifique 10^6 bytes pero también es posible que signifique 2^{20} bytes. Puede depender tanto del fabricante del dispositivo como a qué se esté refiriendo (módulos de memoria RAM, unidades de almacenamiento, hardware de red, ...).

Para hacernos una idea del volumen de datos que maneja la humanidad, según las predicciones el volumen de datos en el mundo se calculaba en unos 4.4 zettabytes en 2013, y tiene un crecimiento exponencial según el cual se espera que pueda llegar a los 163 zettabytes para el año 2025.

¿De dónde vienen todos esos datos?

- Datos de usuarios y/o clientes de instituciones y empresas.
- Datos generados por transacciones (compras, transferencias, ...).
- Datos adquiridos por sensores (de temperatura, de humedad, ...).
- Datos subidos a redes sociales (textos, imágenes, vídeos, ...).
- Datos relacionados con la salud (historiales y pruebas realizadas a pacientes).
- Datos de geolocalización (posicionamiento en cada momento según GPS).
- Datos guardados en logs (de todos los accesos que hacemos a páginas web).
- Datos producidos por el Internet de las cosas (de los diversos dispositivos IoT).
- Datos producidos por la genómica (cada vez que se secuencia un genoma).
- Datos de meteorología (información obtenida por satélites y las predicciones realizadas a partir de la misma).
- Datos producidos por cámaras (imágenes estáticas y vídeos producidos).
- Datos producidos por micrófonos (grabaciones de sonido producidas).
- Datos de RFID (aquellos con los que se tratan al realizar identificación por radiofrecuencia).

- Datos producidos por los sectores energético e industrial (toda la información que se genera alrededor de la energía y la industria).
- Datos Open Data (todos los datos abiertos liberados ya sea a nivel gubernamental o no gubernamental).

Debes conocer

¿A partir de qué cantidad de datos es Big Data?

No existe ninguna entidad u organismo que regule de algún modo cuál es el tamaño de datos concreto a partir de la cual se considera que estamos en un ambiente Big Data.

Simplemente nos quedaremos con que los sistemas para Big Data hoy en día trabajan con volúmenes del orden de los petabytes (PB) e incluso de los exabytes (EB).

Para saber más

En el siguiente enlace puedes ver más información los Datos Abiertos (o Open Data) sobre una de las fuentes de datos más interesantes, ya que son de dominio público.

Datos abiertos

Autoevaluación

Si en algún atributo de la especificación de un dispositivo hardware vemos un valor de 1 kB, ¿a cuántos bytes corresponde?

- ○ A 1000 bytes siempre.
- ○ A 1020 bytes siempre.
- ○ Dependiendo de la situación, quizás se refiera a 1 kB (que corresponde a 1000 bytes), o a 1KiB (que corresponde a 1024 bytes).
- ○ A 1024 bytes siempre.

Incorrecto. No siempre.

Incorrecto. Nunca corresponde a 1020 bytes.

Correcto.

Incorrecto. No siempre.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

1.1.2.- Velocidad.

Lucent_Designs_dinoson20 (Dominio público)

No sólo tratamos con una gran cantidad de datos que hay que almacenar y procesar, sino que tales datos a su vez se siguen produciendo a una gran velocidad.

Para hacernos una idea, se calcula que en el mundo se generan cada 60 segundos:

- 350.000 tweets.
- 300 horas de vídeo subidos a YouTube (más los que se suban a otras plataformas).
- 171 millones de correos electrónicos.
- 330 GBs de información generados por sensores de motores de aviones comerciales.

Si volvemos a revisar la enumeración de posibles fuentes de las que provienen los datos que vimos en el apartado anterior, y tenemos en cuenta que en el mundo hay del orden de 7870 millones de personas (datos de 2022), podremos seguir haciéndonos una idea de la gran velocidad a la que todos esos datos se siguen generando cada minuto que pasa.

El problema con respecto a la velocidad no es únicamente el hecho de que el volumen de datos continúe creciendo sin parar (ya que si hemos dimensionado el almacenamiento para el doble de lo que necesitamos entonces aún quedará mucho tiempo para que tal tamaño de almacenamiento sea un problema), sino lo rápido que es necesario obtenerlos y ser capaces de integrarlos junto con los que ya tenemos.

De la gran velocidad a la que llegan datos nuevos nacen las estrategias de procesamiento tipo streaming, las cuales estudiaremos más adelante.

Reflexiona

Si quisiésemos almacenar diariamente un valor de 4 bytes con el peso de los aproximadamente 7870 millones de personas, ¿cuánto nos ocuparía tal información a largo de un año?

Mostrar retroalimentación

$4 * (7{,}890 * 10^9) * 365 = 11{,}5194 * 10^{12}$ bytes = 11,5194 TB

1.1.3.- Variedad.

Además de tener que procesar una gran cantidad de datos que se generan cada vez más rápido, existe el problema añadido de la gran variedad existente en cuanto a la representación de tal información.

Datos estructurados

Los existentes en registros (filas) de bases de datos (típicamente relacionales), los cuales existen dentro de tablas con un esquema definido que nos indica de qué tipo de datos es cada una de las columnas (entero, decimal, textual, fecha, ...).

Algunas de las características de los datos estructurados incluyen:

- Existe una estructura bien definida y organizada.
- Se puede almacenar en tablas, normalmente dentro de columnas verticales y filas horizontales.
- El contenido y formato de los datos está documentado.
- Está organizado en archivos, registros y campos.
- Se puede buscar, ordenar y consultar.
- Los controles de entrada pueden reducir la posibilidad de datos no válidos.

Datos no estructurados

Aquellos que no están regidos por un esquema. Por ejemplo:

- Vídeos.
- Imágenes.
- Audios.

Hay que tener que la proporción de datos en el mundo que son no estructurados se estima en más de un 80% del total, lo cual es fácilmente comprensible teniendo en cuenta la naturaleza de los mismos. Sólo hay que comparar el espacio de almacenamiento que ocupa un vídeo (típicamente varios megabytes) con el que ocupa un registro en una base de datos (típicamente varios bytes).

Hay muchas fuentes de datos no estructurados en Internet, en forma de texto, imágenes, vídeos y audio. Los blogs y foros web públicos también generan datos. Las redes sociales como YouTube, Facebook, mensajería instantánea, RSS y Twitter/X se suman a los datos que se encuentran en Internet. La mayoría de estos datos no están estructurados, lo que significa que no es fácil categorizarlos en una base de datos sin algún tipo de procesamiento. Los datos que no están estructurados se pueden extraer y procesar de varias maneras:

- Bases de datos NoSQL y lagos de datos (Data Lakes).
- Web Scrapping (Raspado Web).
- Interfaces de programas de aplicaciones (APIs).

Datos semiestructurados

Son datos definidos según una cierta estructura pero que no tienen naturaleza relacional (es decir, no son registros de una tabla con un esquema determinado).

Por lo general se almacena en ficheros de texto siguiendo un cierto formato

preestablecido, de modo que se mantiene la flexibilidad que ofrece el fichero (para poder almacenar lo que sea necesario) a la vez que es posible determinar qué significa cada una de las porciones de información que se encuentran dentro del mismo.

Ejemplos de formato de fichero en los que se guardan datos semiestructurados:

- CSV.
- XML.
- JSON.

Metadatos

Los metadatos son datos extra (muchas veces generados de forma automática) que se guardan acerca de los propios datos para favorecer su interpretabilidad posterior.

Ejemplos de metadatos que pueden acompañar a los datos convencionales son:

- Información extra sobre su estructura.
- Fuente.
- Autor.
- Fecha de creación.
- Resolución en pixels (si se trata de una imagen o un vídeo).
- Duración (si se trata de un vídeo).
- Frecuencia de muestreo (si se trata de un audio).
- Tipo de compresión.

Autoevaluación

¿A qué tipo de información corresponde, generalmente, un fichero con extensión .json?

Sugerencia

- ○ Estructurados.
- ○ No estructurados.
- ○ Semiestructurados.
- ○ Metadatos.

Incorrecto. Los datos estructurados son por ejemplo los de las bases de datos relacionales.

Incorrecto. En un JSON hay algo de estructura.

Correcto.

Incorrecto. Los metadatos son datos acerca de los datos.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

1.1.4.- Veracidad.

Un problema extra con el que tenemos que tratar es el hecho de que los datos no siempre cuentan con la calidad deseada o no son totalmente fieles a la realidad.

Este término está muy relacionado con el concepto de relación señal/ruido en cualquier flujo de información.

- El ruido son datos que no pueden ser convertidos en información (ya sea porque no la contienen o porque ésta está corrupta y es irrecuperable).
- La señal está constituída por datos que sí pueden ser convertidos en información con sentido.

Para saber más

En el siguiente enlace de la Wikipedia puedes saber más sobre lo que significa la relación señal/ruido:

Relación señal/ruido

Por ello, por un lado es necesario conocer en qué condiciones se adquirieron los datos (para poder así estimar su nivel de veracidad), mientras que por otro lado en muchos casos será necesario llevar a cabo un procesamiento específico de los mismos con el fin de resolver posibles problemas y eliminar información inválida.

Por lo general los datos producidos de modo automático (como la generada cuando realizamos transacciones) contienen menos ruido que los que producen personas (como los posts de un blog).

Autoevaluación

¿A qué nos referimos cuando decimos que hay ruido en los datos?

- A que el fichero de audio se grabó con un micrófono de baja calidad.
- A que guardamos el sonido en un ambiente ruidoso.
- A que parte de los datos no contienen información usable o de la que

se pueda obtener algún tipo de valor.

- No puede haber ruido en los datos.

Incorrecto. Ese es otro tipo de ruido.

Incorrecto. Ese es otro tipo de ruido.

Correcto.

Incorrecto. Parte de los datos pueden no contener información usable, y eso recibe el nombre de ruido.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

1.1.5.- Valor.

El concepto de valor en relación a los datos tiene que ver con cómo de útiles son estos para una institución, empresa o persona.

nattanan23 (Dominio público)

Tiene mucho que ver con el concepto de veracidad que ya hemos visto, ya que por lo general cuanto más veraces (fieles a la realidad) sean los datos, más valor se puede obtener de ellso.

También depende en gran medida del tiempo transcurrido desde que se produjeron tales datos. Por ejemplo, si estamos operando en bolsa, el dato que nos indica el valor de una acción es mucho más valioso si corresponde a hace 1 segundo que si corresponde a hace 1 hora. En terminos generales, cuanto más rápido seamos capaces de hacer llegar el dato desde donde se produce al lugar en el que se toman las decisiones, más valor podremos obtener de ellos.

Es también muy importante que los datos sean lo más completos posible para poder producir el valor deseado. Es decir, no sólo que sean veraces (que lo que viene sea correcto) sino que sean completos (que venga todo lo que necesitamos).

Por último, la propia interpretación del dato también juega un papel vital a la hora de poder obtener valor. Por ejemplo sería absurdo estar almacenando un valor de temperatura obtenido por un sensor que está bajo tierra y querer utilizarlo para una científica sobre temperatura ambiente. En este caso el dato podría ser perfectamente veraz pero la interpretación del mismo estaría siendo errónea, lo cual disminuiría el valor producido.

Reflexiona

Si los datos van perdiendo valor con el tiempo y tenemos muchos datos antíguos, ¿merece la pena utilizarlos siempre junto con los más nuevos?

Mostrar retroalimentación

No siempre. Dependerá del caso.

Sobre todo para el análisis descriptivo suele ser útil tener en cuenta datos antíguos para comprobar cuáles son las tendencias (es decir, tener algo con lo que comparar los datos nuevos).

En otros casos, como por ejemplo en la generación de modelos predictivos para intentar acertar con una oferta, lo que querremos es tener cuantos más datos nuevos mejor para así no necesitar usar los más antiguos (ya que los gustos de los usuarios son cambiantes).

1.2.- Qué conseguimos gracias a Big Data.

En esta sección veremos qué nos aporta no sólo el ser capaces de obtener y almacenar con grandes cantidades de datos sino también el poder tratarlos y analizarlos gracias a las metodologías y tecnologías de Big Data.

Aportes generales de Big Data

Las metodologías y tecnologías para Big Data nos permiten realizar diversas operaciones con grandes cantidades de datos, entre las cuales se encuentran:

- Capturarlos desde sus orígenes.
- Integrarlos para poderlos almacenar de un modo unificado.
- Almacenarlos de un modo distribuido y replicado, gracias lo cual conseguimos altos valores de disponibilidad.
- Tratarlos de forma distribuida, empleando para ello un alto número de máquinas que los procesan en paralelo.
- Aplicar técnicas de minería de datos (también llamado ciencia de datos cuando esa minería de datos se realiza en ambientes Big Data) para crear modelos predictivos.
- Usar esos modelos para realizar predicciones a utilizar en sistemas automáticos.
- Crear visualizaciones y cuadros de mando usando tanto los propios datos como los modelos creados para así dar soporte a la toma de decisiones.

El ser capaces de realizar tales operaciones con los datos, nos permiten obtener los siguientes aportes y beneficios (entre otros):

- Generar registros más detallados mediante la integración desde diversas fuentes.
- Optimizar las operaciones de instituciones y empresas.
- Poder actuar de modo inteligente basándonos en la evidencia de los datos.
- Identificar nuevos mercados.
- Realizar predicciones basándonos en modelos creados a partir de los datos.
- Detectar casos de fraude e impagos.
- Dar soporte a la toma de decisiones.
- Realizar descubrimientos científicos.
- Ayudar a los médicos a detectar enfermedades en función del historial de los pacientes y las pruebas que se les realizan.
- Crear nuevos fármacos más efectivos y con menos efectos secundarios.

Autoevaluación

¿Cuáles de los siguientes son posibles beneficios de las metodologías y tecnologías Big Data?

- ☐ Soportar la toma de decisiones.

- [ ] Mejorar las operaciones en empresas e instituciones.
- [ ] Ayudar a detectar enfermedades.
- [ ] Ayudar a los científicos a realizar nuevos descubrimientos.

Mostrar retroalimentación

Solución

1. Correcto
2. Correcto
3. Correcto
4. Correcto

1.2.1.- Desde los eventos al valor.

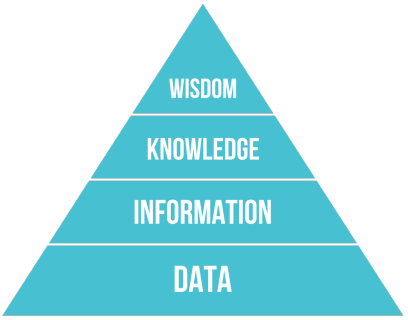
El tratamiento de los datos a lo largo de diversas capas de procesamiento sucesivas nos permite llegar desde los meros eventos que se producen en nuestro mundo hasta la sabiduría que necesitamos para obtener valor gracias a poder tomar las mejores decisiones.

Eventos:

En nuestro mundo se producen eventos constantemente.

- Una estación meteorológica una mediciones de temperatura, humedad, presión atmosférica, etc.
- Una cámara toma imágenes dentro de una fábrica.
- Alguien pide un préstamo.
- Alguien realiza una llamada telefónica.
- Alguien realiza un pago con tarjeta.
- Un hospital realiza una prueba médica a un paciente.
- ...

Pirámide DIKW



Longlivetheux (CC BY-SA)

Datos (Data):

Los eventos son reflejados de algún modo, generándose de ese modo datos que pueden ser almacenados para su uso posterior.

- Registros de bases de datos.
- Ficheros (en diversos posibles formatos).

Ejemplo: Llueve 4 mm (= observación).

Información (Information):

Cuando les damos contexto a los datos, organizándolos de algún modo lógico, tenemos información.

- Distintos registros referentes a pagos con tarjeta quedan almacenados en una misma tabla.
- Usamos jerarquías de carpetas para organizar distintos ficheros en función de su tipo o significado (fotografías, audios, facturas, ...).

Ejemplo: La temperatura bajó y la humedad aumentó a las 10:00 del 10 de octubre en Almería, España.

Conocimiento (Knowledge):

Si tratamos la información dándole un significado, podemos obtener conocimiento.

- A partir de gran cantidad de datos se generan modelos mediante los cuales se representa la realidad y que pueden ser utilizados para realizar predicciones.

Ejemplo: Caída de temperatura + aumento rápido de la humedad + área de presión más baja = lluvia. ← Porque interactúan la evaporación, las zonas de presión, los gradientes de temperatura, los cambios y las precipitaciones.

Sabiduría (Wisdom):

Si una vez tenemos conocimiento en forma de modelos predictivos añadimos el entendimiento necesario para saber de qué modo emplearlos. Como resultado obtenemos sabiduría.

Ejemplo: Podemos anticipar por qué y cuándo lloverá en el futuro según nuestras observaciones y nuestro modelo matemático.

Valor:

La sabiduría de por sí misma no genera ninguna acción. Sin embargo, si realizamos acciones basándonos en la sabiduría, esas acciones serán mejores que las que podamos tomar sin basarnos en los datos.

La diferencia entre el resultado que podemos obtener basándonos en la sabiduría que producen los datos y el que obtendríamos si no los hubiésemos tenido en cuenta para nada, es el valor añadido que conseguimos.

Debes conocer

A pesar de que dentro de las tecnologías de Big Data se suele englobar lo relacionado con obtener valor del dato, en la práctica son la minería de datos o la ciencia de datos las disciplinas que terminan de obtener el valor (haciendo uso de esas tecnologías).

La Minería de Datos es una rama de la Inteligencia Artificial que emplea técnicas de Aprendizaje Automático para obtener valor de los datos.

La Ciencia de Datos en el fondo es misma Minería de Datos pero haciendo énfasis en que se realiza en entornos de Big Data.

Sin embargo, puedes encontrar los términos Minería de Datos y Ciencia de Datos siendo empleados de forma equivalente (Minería de Datos en Big Data y Ciencia de Datos fuera de Big Data).

Autoevaluación

¿Cuáles de los siguientes son eventos susceptibles de generar datos?

- [ ] Un pago con tarjeta.
- [ ] Un alta de usuario en una web.
- [ ] Una medida de presión atmosférica en una estación meteorológica.
- [ ] Un análisis de sangre de un paciente en un hospital.

Mostrar retroalimentación

Solución

1. Correcto
2. Correcto
3. Correcto
4. Correcto

2.- Clusters de computadoras.

Caso práctico

Phool Proof (CC BY-NC-SA)

Quedan 2 minutos escasos para que la clase de física aplicada finalice, y **Simón** no puede dejar de mirar por el gran ventanal del aula. Desde que empezó a trabajar en la universidad no se había sentido tan ilusionado como hoy.

Por fin un camión para en la entrada principal de la universidad.

Ha llegado.

Las puertas traseras del camión se abren y varias personas comienzan a descargar cajas en las que aparece el nombre de una conocida marca de ordenadores.

—¿Qué son esas cajas, don Simón? —pregunta uno de los alumnos.

—¡Son los ordenadores para el nuevo clúster!

En ambientes de computación, un cluster es un conjunto de computadoras (también referenciados como servidores o como nodos) conectados entre sí mediante red para trabajar como una única unidad resolviendo cargas de trabajo de forma conjunta.

Históricamente los clusters se construían utilizando computadoras especializadas muy caras. Sin embargo, más adelante han ido apareciendo diversos frameworks o plataformas de computación distribuída que emplean computadoras de uso común (el llamado commodity hardware), gracias al considerable aumento sus prestaciones.

Para saber más

Ley de Moore:

Según la Ley de Moore, cada aproximadamente 2 años se duplica el número de transistores en los nuevos procesadores que salen a la venta.

Puedes ver más información sobre la Ley de Moore y lo que significa en el siguiente enlace:

Ley de Moore

Para saber más

Puedes ver más información sobre los clusters de computadoras en el siguiente enlace:

Clúster de computadoras

El uso de clústers nos da una serie de ventajas respecto al uso de computadoras de forma individual:

- Alto rendimiento.
- Alta disponibilidad.
- Equilibrado de carga.
- Escalabilidad.

Alto rendimiento:

Dado que cada componente del cluster es una computadora completa, con sus propios recursos (procesdor, memoria y almacenamiento), las cargas de trabajo susceptibles de paralelización pueden acelerarse en gran medida dividiéndolas en subtareas y distribuyéndolas para que sean ejecutadas en los distintos nodos.

Gracias a esto se pueden resolver problemas muy complejos que no sería posible resolver en un tiempo razonable en una máquina individual por muy potente que ésta sea.

Alta disponibilidad:

Mediante una contínua monitorización entre los propios nodos del cluster, se puede detectar la no disponibilidad de un subconjunto de los mismos (ya sea por fallo eléctrico, por avería o por corte de las comunicaciones) y se pueden tomar medidas para que los servicios o datos que hay (o había) en esas máquinas sigan estando disponibles.

- Rearrancando un nodo caído o arrancando un nuevo nodo para suplirlo.
- Respondiendo las peticiones desde otro nodo del clúster que también contenga una réplica de esos datos.

Equilibrado de carga:

El equilibrado de carga (o también balance o balanceo) se consigue mediante algoritmos destinados a distribuir las cargas de trabajo entre los diversos nodos del clúster para así evitar cuellos de botella. Tales cuellos de botella se producen cuando el envío de trabajos a nodos sobrecargados aumenta la latencia media con la que tales trabajos son finalizados.

Para ello, se realiza una monitorización del estado de carga de cada nodo y se decide para cada paquete de trabajo a qué nodo enviarlo, atendiendo a:

- El tamaño del trabajo.
- El estado de carga de cada nodo.
- La potencia de procesamiento de cada nodo.

Escalabilidad:

Gracias a que el clúster está formado por un número indeterminado de nodos, no sólo conseguimos una mayor potencia de cálculo al utilizarlos para una misma tarea, sino que podemos hacer crecer dicha potencia de cálculo añadiendo nuevos nodos. En otras palabras, la potencia de cálculo del clúster es ampliable.

Esta característica es muy desable para sistemas Big Data, ya que desaparece la necesidad de realizar una estimación de potencia necesaria a priori, lo cual en por lo general siempre lleva a una sobreestimación para guardar un margen de seguridad. Con un clúster escalable podemos comenzar con un número determinado de nodos e ir añadiendo más según sea necesario.

Es interesante conocer la diferencia entre escalado horizontal y vertical:

- **Escalado vertical (scale-in):**
  - Es el que se consigue mejorando las características hardware de la computadora (individual) en el que se están ejecuntando las cargas de trabajo (procesador, memoria o almacenamiento). Por lo tanto, está limitado por la mejor especificación de hardware que sea posible encontrar en el mercado.
  - Por ello, aunque reciba el nombre de "escalado" en la práctica no sirve para conseguir la característica de escalabilidad.
- **Escalado horizontal (scale-out):**
  - Es el que se consigue añadiendo más nodos a un clúster.
  - Por ello es el tipo de escalado que realmente nos permite conseguir la característica de escalabilidad.

Para saber más

En los siguientes enlaces puedes ver más información sobre lo que significa alto rendimiento, alta disponibilidad, equilibrado de carga y escalabilidad:

Clúster de alto rendimiento

Clúster de alta disponibilidad

Equilibrio de carga

Escalabilidad

Autoevaluación

¿Qué hacemos si un clúster necesita más capacidad de almacenamiento?

- ○ Hacemos escalado vertical de todos los nodos, aumentando el tamaño de almacenamiento de cada uno.
- ○ Hacemos escalado horizontal, añadiendo mas nodos al clúster.
- ○ Hacemos escalado en diagonal, aumentando el almacenamiento en los nodos que tengan menos espacio disponible.
- ○ Hacemos escalado vertical, añadiendo mas nodos al clúster.

De ese modo podríamos tener más almacenamiento, pero no es la filosofía empleada en clústers para Big Data.

Correcto.

El escalado en diagonal no existe en relación a clústers de máquinas.

No, eso se llama escalado horizontal.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto
4. Incorrecto

3.- Conceptos de almacenamiento de datos.

Caso práctico

—¡No hay manera! —dice **Pedro**.

—¿Cómo no va a haber manera? —replica **María**.

—Te digo que no hay manera. La tabla que necesitamos tiene tantos campos y tantos registros que no nos cabe ni en el disco duro del ordenador. ¡Y es el más grande que se puede comprar ahora mismo!

IT-STUDIO (Dominio público)

—¿Y entonces?

—Tendríamos que partir la tabla y guardar los fragmentos en datos en varios ordenadores.

—Claro, ¿y luego cuando tengamos que hacer una consulta cómo sabemos en qué ordenador está justo lo que buscamos? ¿Y si está esparcido por todos los fragmentos?

En esta sección vamos a realizar un recorrido a lo largo de una serie de conceptos relacionados con almacenamiento que es importante conocer si vamos a trabajar en entornos Big Data.

Veremos aquí un esquema/resumen para que puedas tener una vista general:

- **Base de Datos Relacional:**
  El tipo de bases de datos más utilizado en el mundo (pero no escalable para Big Data).

- **Dataset:**
  Un conjunto de datos (quizás enorme).

- **Almacén de Datos:**
  Una sistema especial para almacenar datos (típicamente para analítica).

- **Data lake:**
  Un data lake o lago de datos es un repositorio de almacenamiento que contiene

grandes cantidades de datos en formatos nativos y sin procesar.

- **ACID:**
  Una serie de propiedades que deben cumplir las bases de datos que vayan a ser usadas para realizar transacciones.

- **Teorema CAP:**
  Un teorema acerca de las propiedades que podemos conseguir en una base de datos distribuida.

- **BASE:**
  Un principio de diseño de base de datos distribuidas.

3.1.- Base de Datos Relacional.

Debes conocer

El lenguaje comúnmente empleado para interactuar con bases de datos relacionales es SQL. Puedes ver información sobre SQL en el siguiente enlace:

SQL

Una base de datos relacional es un almacén de información que almacena registros dentro tablas.

Dichas tablas constan de filas (una por cada registro) y de columnas (los atributos de los que está compuesto cada registro).

Para cada tabla se define un esquema, en el cual se indica qué atributos tienen los registros de la misma y de qué tipo son (entero, decimal, texto, fecha, ...).

Gracias a la uniformidad de los datos que hay dentro de cada tabla, los motores de bases de datos relacionales pueden ofrecer un altísimo rendimiento a la hora de realizar búsquedas. Tales búsquedas pueden realizarse dentro de una única tabla o incluso afectando a varias tablas a la vez, a través de las relaciones existentes entre las mismas (de ahí "base de datos relacional").

Una de las características clave de las bases de datos relacionales para su alto rendimiento es su capacidad para generar índices sobre columnas de las tabla, gracias a los cuales se acelera tanto la búsqueda dentro de una tabla en particular como en enlazado de registros de distintas tablas que están relacionados.

Para saber más

En el siguiente enlace puedes ver más información sobre lo que es un sistema de gestión de bases de datos relacionales:

RDBMS

Reflexiona

Las Bases de Datos Relacionales ofrecen un alto rendimiento para realizar transacciones, pero sus motores no están pensados para el caso en el que una tabla sea tan grande que todos sus registros no puedan ser almacenados dentro de un mismo servidor.

De modo que, ¿son las Bases de Datos Relacionales apropiadas para entornos Big Data?

Mostrar retroalimentación

No. Las bases de datos relacionales escalan en vertical (esto es, para tener más potencia o capacidad de almacenamiento ponemos un servidor mejor).

Esto cual supone un cuello de botella para si nos encontramos en un entorno Big Data, ya que no siempre podremos poner un servidor con CPU más potente, con más memoria o con más capacidad de almacenamiento.

Autoevaluación

¿Cuál de las siguientes afirmaciones es cierta en relación a las bases de datos relacionales?

- Utilizan MySQL como lenguaje de consulta.
- No es necesario conocer los tipos de datos que se van a almacenar desde un primer momento, sino que se determina al realizar su lectura.
- Podemos utilizar el tipo de datos RDBMS, en el cual cabe cualquier número de bytes ya que se guarda en ficheros espefícos fuera de la base de datos.
- Si creamos índices para las columnas sobre las que vayamos a hacer búsquedas, éstas se ejecutarán más rápido.

Incorrecto. El lenguaje empleado es SQL.

Incorrecto. Antes de comenzar a insertar datos debe determinarse el

esquema de cada tabla y con ello qué tipo de datos corresponde a cada columna.

Incorrecto. RDBMS significa Relational Database Management System, y tiene que ver con el motor de la base de datos.

Correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

3.2.- Dataset.

Un dataset es una colección de datos que guardan una cierta relación debido a la cual tiene sentido tratarlos juntos.

Ejemplos de dataset son:

- Una colección de tweets.
- Una colección de posts.
- Una colección de imágenes.
- Una serie de registros de base de datos relacional.
- Una serie de registros de base de datos no relacional.
- Una sucesión de medidas de una estación metereológica.

Tal dataset a su vez puede estar almacenado en diversos formatos:

- Ficheros de texto plano (CSV, XML, JSON, o sin formato en particular).
- Tablas de base de datos.
- Ficheros multimedia (imagen, vídeo, audio, ...).

Para saber más

Puedes ver más información sobre lo que es un dataset (en castellano "conjunto de datos") en el siguiente enlace:

Conjunto de datos

Autoevaluación

¿Cuál de las siguientes afirmaciones es correcta respecto de un dataset?

- ○ Siempre vienen en ficheros de texto plano.
- ○ Contienen datos de usuarios.
- ○ No pueden contener datos de usuarios porque constituye un uso prohibido.
- ○ Que contenga imágenes no significa que no pueda contener también texto, audio o vídeo.

Incorrecto. Pueden venir en distintos formatos.

Incorrecto. Los datos de usuario sólo son un posible caso dentro de las múltiples naturalezas que puede tener un dataset.

Incorrecto. Pueden contener datos de usuarios y tal uso no está prohibido siempre que se cumpla con las normativas.

Correcto. Muchos datasets contienen un único tipo de información, pero por poder pueden contener cualquier combinación si eso es interesante para la tarea que se va a realizar con esos datos.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

3.3.- Almacén de Datos.

Un almacén de datos (del inglés, data warehouse) es un repositorio central de datos a nivel institucional o empresarial, dentro del cual se almacenan tanto datos actuales como históricos.

Se emplean tanto para inteligencia de negocio (BI) como para realizar consultas analíticas, por lo que además de tablas relacionales también suelen incluir en su interior subsistemas de tipo OLAP.

Por lo general, los datos que contienen son cargados periódicamente desde sus fuentes (por ejemplo sistemas SCM, ERP o CRM) mediante procesos de tipo ETL. Esto significa que la información que contienen es por general una instantánea del estado de los datos a cierta fecha, por lo que se trata de un almacenamiento que por lo general no se va a emplear para el uso de transacciones (sino para inteligencia de negocio o analítica, la cual es su función).

Para saber más

En el siguiente enlace puedes ver más información sobre lo que es un almacén de datos.

Almacén de datos

Autoevaluación

¿Un almacén de datos puede incluir en su interior una base de datos relacional?

- ○ Sí.
- ○ No, sólo puede incluír subsistemas OLAP.
- ○ No, sólo bases de datos de tipo NoSQL.

Correcto.

Incorrecto. Suelen incluir subsistemas OLAP, pero también pueden incluir otros tipos de almacenamiento.

Incorrecto. Puede incluir diversos tipos de almacenamiento.

Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto

3.4.- Data lake.

¿Qué es Data Lake?

Un *Data lake* es un repositorio de almacenamiento que contiene una gran cantidad de datos en bruto y que se mantienen allí hasta que sea necesario. A diferencia de un *Data warehouse* jerárquico que almacena datos en ficheros o carpetas, un *Data lake* utiliza una arquitectura plana para almacenar los datos.

Este repositorio no tiene límite de tamaño y en él se pueden almacenar datos de tres tipologías distintas:

- **Estructurados.** Son aquellos que tienen un formato estandarizado, con patrones claramente predefinidos. En esta categoría se enmarcan los archivos de Excel, los datos de control de inventario o los resultados de los formularios web, entre otros.
- **No estructurados.** Son aquellos que no tienen un formato definido, es decir, no tienen una estructura uniforme. Este tipo de datos son los más abundantes. Los vídeos, las imágenes, los audios, los correos electrónicos o los contratos son algunos ejemplos que conforman esta categoría.
- **Semiestructurados.** Son aquellos que, pese a tener un formato definido, no resultan fácilmente comprensibles. En esta categoría se incluyen las etiquetas de lenguaje HTML, los correos electrónicos o los gráficos, por ejemplo.

A cada elemento de un *Data lake* se le asigna un identificador único y se etiqueta con un conjunto de etiquetas de metadatos extendidas. Cuando se presenta una cuestión que debe ser resuelta, podemos solicitarle al *Data lake* los datos que estén relacionados con esa cuestión. Una vez obtenidos podemos analizar ese conjunto de datos más pequeño para ayudar a obtener una respuesta.

Para manipular e interpretar eficazmente los datos almacenados, no obstante, es importante implementar una arquitectura que reúna los siguientes componentes clave:

- **Ingesta de datos.** Deben contar con un sistema de capas de ingesta que sea fácilmente escalable y que pueda extraer datos de diversas fuentes, ser capaces de procesar datos tanto en tiempo real como por lotes y poder admitir cualquier tipo de dato, independientemente de su naturaleza.
- **Almacenamiento de datos.** El sistema debe ser capaz de almacenar y tratar grandes volúmenes de datos sin procesar y de soportar sistemas de cifrado y compresión de datos.
- **Seguridad de datos.** El sistema debe ofrecer la máxima seguridad, independientemente del tipo de datos que almacenen.
- **Analítica de datos.** Los datos almacenados en los 'data lakes' deben poder analizarse de forma ágil y eficiente a través de herramientas de análisis de datos o del propio machine learning a fin de extraer información de interés.
- **Gobierno de datos.** Todo el proceso de ingesta, preparación, categorización, integración y disponibilidad de los datos debe estar acompañado de un modelo de gobierno que facilite entender qué significan los datos, qué calidad tienen, dónde y cuándo están disponibles y finalmente quién los puede consultar. Además, este modelo debe garantizar un seguimiento de todos los cambios que se produzcan en el ciclo de vida de los datos.

El *Data lake* se asocia a menudo con el almacenamiento de objetos orientado a *Hadoop*. En este escenario, los datos de una organización se cargan primero en la plataforma *Hadoop* y, a continuación, se aplican las herramientas de análisis y de minería de datos a

los datos que residen en los nodos clúster de *Hadoop*.

Cada vez más el término está siendo aceptado como una forma de describir cualquier gran conjunto de datos en el que el esquema y los requisitos de datos no se definen hasta que los datos se consultan.

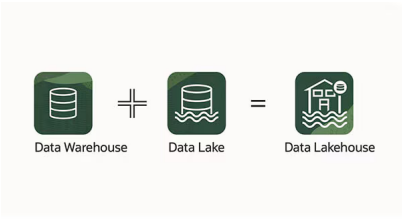
Principales diferencias entre *Data Lakes* y *Data Warehouses*

Los *Data lakes* suelen emplearse de manera conjunta con otro sistema que permite el almacenamiento y procesamiento de grandes volúmenes de datos: los *Data Warehouses*. Estos dos repositorios guardan importantes semejanzas, en el sentido de que los dos se emplean para recopilar datos, pero entre ambos existen diferencias que conviene conocer:

- **Naturaleza de los datos.** Los *Data lakes* pueden recopilar todo tipo de datos, independientemente de su naturaleza, mientras que los *Data warahouses* sólo almacenan datos estructurados.
- **Formato de los datos.** Los *Data lakes* almacenan datos en crudo, es decir, con sus atributos originales, mientras que los 'data warehouses' almacenan datos ya procesados.
- **Fuente de los datos.** Mientras que la información de los *Data lakes* proceden del Big Data, el internet de las cosas, las redes sociales o los datos de las plataformas de streaming; los *Data warahouses* se alimentan de datos de aplicaciones, negocios, transacciones o reportes.
- **Escalabilidad.** Los *Data lakes* pueden escalar de manera sencilla y a un bajo coste, mientras que la escalabilidad de los *Data warahouses* es más compleja.
- **Usos.** Los datos recopilados por los *Data lakes* pueden emplearse para realizar análisis predictivo o en tiempo real, así como para alimentar los algoritmos machine learning, mientras que los *Data warahouses* pueden emplearse para realizar informes o para sustentar la inteligencia de negocios.

Debes conocer

Un ***Data lakehouse*** es una arquitectura de datos que combina un *Data lake* y un almacén de datos. Los *Data lakehouses* permiten el aprendizaje automático, la inteligencia empresarial y las estadísticas predictivas, lo que permite que las organizaciones aprovechen el almacenamiento flexible y de bajo costo para todo tipo de datos estructurados, no estructurados y semiestructurados, a la vez que proporciona estructuras de datos y funciones de administración de datos.



A continuación podemos ver algunos ejemplos de implementaciones de *Data lakehouses*:

- Delta Lake
- Databricks

- Snowflake
- Microsoft Fabric
- Google BigLake

3.5.- ACID.

ACID es el principio fundamental de diseño por el cual se rigen las bases de datos que se crean para uso transaccional.

Está formado por 4 características de obligado cumplimiento, correspondiendo cada una de ellas a una de las letras del acrónimo:

- Atomicidad (**A**tomicity).
- Consistencia (**C**onsistency).
- Aislamiento (**I**solation).
- Durabilidad (**D**urability).

Este tipo de gestión realiza un control pesimista de la concurrencia, dando por hecho que cualquier problema que pueda ocurrir ocurrirá tarde o temprano por poco probable que sea (aplicando la Ley de Murphy).

Para conseguirlo, el motor de la base de datos bloquea registros individuales e incluso tablas completas en determinados momentos para asegurar que la consistencia se mantiene en todo momento.

Atomicidad:

La atomicidad implica que las operaciones realizadas sobre la base de datos o bien tienen éxito o fallan por completo (en tal caso dejando la base de datos exactamente como estaba antes de comenzar la operación).

Por ejemplo, si una transacción consiste en insertar dos registros y la primera inserción es exitosa pero la segunda falla (por ejemplo porque el formato de un atributo no cumple con el esquema de la tabla), el motor de la base de datos no consolida la primera inserción.

Consistencia:

La consistencia nos asegura que la base de datos siempre es vista desde fuera en un estado consistente, comprobando siempre que los datos cumplen con los esquemas y restricciones de las tablas antes de escribirlos en ellas.

Gracias a ello, cualquier base de datos en estado consistente sigue en estado consistente tras una transaccion exitosa.

Aislamiento:

El aislamiento nos asegura que los resultados de una transacción no son visibles por otras operaciones hasta que tal transacción haya sido completada.

Esto significa que si una transacción consiste en insertar 2 registros, ningún

otro usuario o proceso podrá hacer una selección en la que aparezca sólo el primero de ellos (verá ambos si la transacción ha finalizado, o ninguno si aún no ha terminado).

Durabilidad:

La durabilidad nos asegura que los resultados de las escrituras (inserciones o actualizaciones) en la base de datos son permanentes. Esto implica que tales escrituras no se pierdan en el caso de que la máquina se apague tras realizar la transacción, lo cual se consigue persistiendo (guardando) la información en un sistema de almacenamiento no volátil. Quedaría por lo tanto descartada una base de datos que mantuviese la información úncamente en memoria.

Es importante tener en cuenta que lo accesos a almacenamiento no volátiles son alrededor de 2 órdenes de magnitud (es decir, unas 100 veces) más lentos que los accesos a memoria, razón por la cual el hecho de mantener la característica de durabilidad es un limitante para la velocidad a la que pueden operar las bases de datos que cumplen con ACID.

Reflexiona

Las bases de datos relacionales se usan en el día a día para operaciones transaccionales.

¿Son, por lo tanto ACID?

Mostrar retroalimentación

La pregunta está mal formulada. El hecho de que se usen para operaciones transaccionales no implica que sean ACID.

La pregunta correcta sería si cuando vamos a usar una base de datos para operaciones transaccionales debemos asegurarnos de que es (es decir, que cumple con) ACID. Y la respuesta a esa pregunta sería que sí.

Para saber más

Puedes ver más información sobre ACID en el siguiente enlace:

ACID

3.6.- Teorema CAP.

El teorema CAP (también conocido como conjetura de Brewer) establece que una base de datos distribuida sólo puede cumplir como máximo con 2 de las siguientes 3 propiedades:

- Consistencia (**C**onsistency).
- Disponibilidad (**A**vailability).
- Tolerancia a particionamiento (**P**artition tolerance).

En otras palabras, según el teorema, nunca puede cumplirse C+A+P, sino que habrá que escoger siempre entre C+A, C+P o A+P a la hora de diseñar la base de datos distribuída.

Esto también implica que a la hora de seleccionar una base de datos distribuída tendremos en primer lugar que decidir de cuál de las 3 características estamos dispuestos a prescindir (C o A o P), y asegurarnos de que la base de datos cumple con las 2 características de las cuales no prescindimos.

Consistencia:

Cualquier lectura realizada (independientemente de sobre qué nodo se produzca) siempre muestra el estado posterior a la última escritura realizada (sobre cualquier nodo) o un error. Es decir, la base de datos tiene permitido devolver un error si no puede devolver estado más actual, pero en ningún caso puede devolver un estado ya desfasado.

Disponibilidad:

Toda petición recibe una respuesta no errónea, sin la garantía de que el estado observado sea el correspondiente a la última escritura en algún nodo de la base de datos.

Tolerancia al particionamiento:

El sistema sigue funcionando y produciendo respuestas aún en el caso de que se haya perdido la comunicación con/entre algunos nodos, lo cual implica que se pueden recibir lecturas desde unos nodos que no incluyan información escrita en otros.

Para mostrar la razón por la que sólo 2 de las 3 propiedades del teorema CAP pueden cumplirse a la vez en una de base de datos distribuída, veremos los siguientes escenarios:

- Si se requiere consistencia (C) y disponibilidad (A), los nodos necesitan estar comunicados para asegurar la consistencia y poder devolver siempre respuestas que no sean de error. Por lo tanto asegurar la tolerancia al particionamiento (P) no es posible.

- Si se requiere consistencia (C) y tolerancia al particionamiento (P), los nodos no pueden mantenerse disponibles (A) durante el tiempo necesario hasta que termine el particionamiento y vuelva a alcancarse un estado consistente (C) entre ellos.
- Si se requiere disponibilidad (A) y tolerancia al particionamiento (P), entonces la consistencia (C) no es posible debido a la necesidad de comunicación entre nodos para que ésta se mantenga. En otras palabras, si se quiere mantener disponible la base de datos en momentos de particionamiento, es obligatorio aceptar lecturas inconsistentes.

Reflexiona

Aunque los cortes de comunicación entre nodos son poco frecuentes, lo cierto es que pueden ocurrir en cualquier momento, y por lo general ninguna institución ni empresa está dispuesta a que su base de datos distribuída deje de funcionar durante esos momentos. En esa gran cantidad de casos en los que se quiere cumplir con la tolerancia a particionamiento (P), ¿qué opciones tenemos?

Mostrar retroalimentación

En tal caso sólo podemos escoger entre cumplir C+P o cumplir A+P, ya que como el teorema CAP establece, las 3 propiedades no son posibles a la vez.

Es decir, tendremos que prescindir de C o de A, decisión de dependerá de los requisitos de uso del sistema.

Para saber más

En el siguiente enlace podrás ver más información acerca del teorema CAP.

Teorema CAP

3.7.- BASE.

BASE es un principio de diseño de bases de datos basado en las restricciones impuestas por el teorema CAP, y típicamente empleado por muchas implementaciones de bases de datos distribuídas.

El significado del acrónimo es:

- Básicamente disponible (**BA**sically available).
- Estado blando (**S**oft state).
- Consistencia eventual (**E**ventual consistency).

Una base de datos que conforme a la filosofía BASE prefiere la disponibilidad antes que la consistencia (es decir, desde el punto de vista del teorma CAP es A+P).

Básicamente disponible:

Significa que la base de datos siempre responde a las solicitudes recibidas, ya sea con una respuesta exitosa o con una noticación de error, aún en el caso de que se produzca particionamiento entre los nodos (que algunos de ellos caigan o no están accesibles mediante red). En ocasiones eso puede significar recibir lecturas desde nodos que no han recibido la última escritura, por lo que el resultado puede no ser consistente.

Estado blando:

Implica que la base de datos puede encontrarse en un estado inconsistente cuando se produce una lectura, de modo que podemos realizar dos veces la misma lectura y obtener dos resultados distintos a pesar de que no haya habido ninguna escritura entre ambas.

En otras palabras, en cada momento sólo tenemos cierta probabilidad de estar viendo el estado final de la base de datos, porque puede haber escrituras que aún no se hayan consolidado en el nodo sobre el que se realiza la lectura.

Eventualmente consistente:

Significa que tras cada escritura, la consistencia de la base de datos sólo se alcanza una vez el cambio ha sido propagado a todos los modos (de ahí que la consistencia sea eventual en lugar de segura). Durante el tiempo que tarda en producirse la consistencia, observamos un estado blando de la base de datos.

Reflexiona

Dado que las bases de datos distribuídas que emplean la filosofía BASE dan prioridad a la disponibilidad a costa de la consistencia, ¿son una buena elección para uso transaccional?

Mostrar retroalimentación

No, ya que para usos transaccionales la consistencia es una característica obligatoria.

Para saber más

Puedes ver más información sobre la filosofía BASE en el siguiente enlace. Ten en cuenta que está en inglés ya que no estía una versión en castellano en el momento de crear este curso.

Eventual consistency (en inglés)

4.- Conceptos de procesamiento de datos.

Caso práctico

Evan-Amos (Dominio público)

En 1995 **Sega** lanzó la **Sega Saturn** (equipada con 2 procesadores Hitachi SH2 de 32 bits a 28.6 MHz) para competir con la **Sony PlayStation** (equipada con un único procesador de 32 bits a 33.8 MHz).

Sobre el papel parecía que la Sega Saturn tenía toda la ventaja en términos de hardware gracias a la suma de sus dos procesadores, que quizás podrían tener un rendimiento equivalente a 28.6 * 2 = 57.2 Mhz, muy por encima de la velocidad de su rival.

Sin embargo fue precisamente esta arquitectura multiprocesador una de las principales causas de su fracaso.

Por un lado, el contar con dos procesadores hizo mucho más complejo para los desarrolladores el crear juegos para ella. En lugar de estar únicamente concentrados en crear grandes títulos, tenían que dedicar parte del tiempo a encontrar el modo distribuir el trabajo entre ambos procesadores, lo cual no era tarea sencilla y además producía errores de software extra que debido a la ejecución en paralelo eran muy complicados de detectar. Por esa razón muchos estudios optaban por desarrollar el juego empleando un único procesador.

Por otro lado, incluso encontrando el modo de paralelizar el juego, el rendimiento final no era el equivalente al de un único procesador el doble de rápido ni mucho menos. Ello se debía a que era imposible tener siempre ambos procesadores a pleno rendimiento (ya que no había modo de conseguir un balanceo de carga perfecto), y aparte en muchas ocasiones uno de ellos tenía que quedar parado a la espera de los resultados del otro. Además había que añadir la sobrecarga extra que conlleva la propia comunicación de información entre los procesadores.

Por último, era mucho más caro producir una videoconsola con 2 procesadores (que eran comprados a Hitachi) que si hubiese tenido sólo uno.

En esta sección vamos a realizar un recorrido a lo largo de una serie de conceptos relacionados con procesamiento de datos de que es importante conocer si vamos a

trabajar en entornos Big Data.

Veremos aquí un esquema/resumen para que puedas tener una vista general:

- **Procesamiento en paralelo:**
  Distintos procesos dentro del mismo procesador.

- **Procesamiento distribuido:**
  Distintos procesos para un mismo trabajo ejecutándose en distintas máquinas.

- **Estrategias de procesamiento de datos:**
  Cómo trabajamos con datos según el tipo de actividad que vayamos a realizar.

- **OLTP:**
  Procesamiento transaccional.

- **OLAP:**
  Procesamiento para analítica.

- **Principio SCV:**
  Un principio que nos dice qué propiedades podemos conseguir en un sistema de procesamiento distribuido.

4.1.- Procesamiento en paralelo.

El procesamiento en paralelo tiene que ver con la capacidad de los sistemas operativos modernos (multitarea) de realizar varias tareas al mismo tiempo.

Los sistemas operativos multitarea existen desde mucho tiempo antes de que comenzasen a aparecer procesadores multihilo (con capacidad hardware para ejecutar varios hilos de forma concurrente) o las placas base multiprocesador (con capacidad para instalar varios procesadores). Para ello, cuentan con un gestor de procesos que se encarga de repartir el tiempo de ejecución entre los diversos procesos que estén ejecutándose en el sistema (en ventanas de tiempo de unos cuantos milisegundos).

Gracias a la aparición de placas base multiprocesador (que datan de tiempos previos a la aparición de los procesadores multihilo), los sistemas operativos instalados en las máquinas equipadas con una de tales placas pudieron tener hardware disponible como para poder realizar más de una tarea realmente al mismo tiempo.

Más adelante, con la aparición de los procesadores multihilo y (algo después) de los procesadores multinúcleo (con varios núcleos que a su vez por lo general también son multihilo), por fin la multitarea real pudo democratizarse para llegar al usuario convencional.

Multinúcleo:

El procesador contiene varios núcleos, cada uno de ellos con una CPU completa.

Multihilo:

Caso en el que una CPU está diseñada de modo que es capaz de atender a más de un hilo de ejecución (por lo general 2) permitiendo al segundo utilizar recursos que en ese momento no esté utilizando el primero.

Por ejemplo, el segundo proceso puede realizar una multiplicación de dos valores mientras el primero no está utilizando el recurso necesario para multiplicar porque se encuentra cargando un dato desde memoria.

Debido a que en ocasiones ambos procesos necesitan usar el mismo recurso (por ejemplo si ambos necesitan multiplicar en un determinado momento), habrá fracciones de tiempo en los que uno de ellos queda parado a la espera de que el otro termine de utlizar el recurso.

Por esa razón, un procesador comercializado como "de 2 núcleos y 4 hilos" no llega a ser equivalente a un procesador de 4 núcleos físicos.

Cuando las tareas que se están ejecutando son independientes (por ejemplo reproducir un audio, grabar vídeo con una webcam y ejecutar un editor de textos), no existe ningún problema de paralelización entre ellas.

El problema en cuanto a la paralelización aparece cuando tenemos una tarea muy compleja (por ejemplo un análisis sobre una gran cantidad de datos, lo cual es el típico ejemplo de trabajo en ambientes Big Data), y necesitamos dividirla en distintas subtareas **independientes** (lo cual no siempre es posible, y aun siendo posible en ocasiones es muy complicado).

Tarea paralelizable:

Si nos piden sumar mil billones de números aprovechando la potencia de un procesador multinúcleo, podemos separar esos números en tantos paquetes como núcleos y ejecutar un proceso para cada uno de ellos.

Cada proceso realiza la suma de los números del paquete recibido y le devuelve el resultado a otro proceso al que ya sólo le queda sumar esos resultados parciales para obtener el resultado final.

Tarea no paralelizable:

Imaginemos que nos piden realizar la siguiente operación con mil billones de números (también aprovechando el procesador multinúcleo):

- Comienza con un resultado de valor 0.
- Mientras queden números, toma el siguiente y:
  - Si valor actual del resultado es par, súmale el número.
  - Si valor actual del resultado es impar, réstale el número.

Esta tarea presenta un evidente problema si queremos paralelizarla, ya que para cada nuevo paso necesitamos conocer el resultado parcial que se ha obtenido hasta el paso anterior.

Esto significa que no hay un modo eficiente de repartir el trabajo entre los núcleos. Aún separando los números en bloques consecutivos, el proceso que recibe el segundo bloque necesitará esperar a que el que recibe el primer bloque produzca su resultado, el que recibe el tercero al que recibe el segundo, y así sucesívamente.

Reflexiona

Pero entonces, si sólo tenemos un único procesador que no es multihilo y el gestor de procesos del sistema operativo reparte tiempos entre los procesos, ¿podemos decir que está realizando varias tareas al mismo tiempo?

Mostrar retroalimentación

No. En este caso es una multitarea simulada, en el sentido de que gracias a que las ventanas temporales son de milisegundos, el usuario humano tiene la impresión de que su ordenador está haciendo varias cosas a la vez (aunque realmente no sea así).

Autoevaluación

¿Cuál de las siguientes afirmaciones es correcta en relación a paralelización de tareas?

- ○ Todas las tareas pueden paralelizarse de modo que se ejecuten más rápido que sin paralelizar.
- ○ No todas las tareas pueden paralelizarse.
- ○ El mayore problema de la paralelización es el tiempo que se tarda en integrar los datos resultantes de tratar cada fragmento.

Incorrecto. En algunos casos es imposible.

Correcto.

Incorrecto. En algunos casos tal integración es prácticamente inmediata o incluso innecesaria.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto

4.2.- Procesamiento distribuido.

El procesamiento distribuido está muy relacionado con el procesamiento en paralelo, con la distinción de que en este caso el procesamento se lleva a cabo en distintas máquinas que se comunican mediante red formando un clúster.

A su vez, cada una de las máquinas del cluster por lo general contará con un procesador multinúcleo, de modo que el procesamiento puede realizarse distribuído en nodos y a su vez en paralelo en procesador. Esto añade una complejidad extra al sistema a la hora de determinar el modo eficiente de aprovechar este doble nivel de paralelismo. Afortunadamente, gracias a los frameworks para Big Data existentes en el mercado, esta gestión se realiza de forma automática y por lo tanto transparente para el desarrollador.

Llegados a este punto, es importante tener en cuenta que para realizar su trababajo de forma conjunta, en muchas ocasiones es necesario que se produzca algún tipo de comunicación entre los distintos procesos que se están ejecutando en paralelo (muchas veces en la forma de un proceso entregando/enviando a otro un conjunto de datos con resultados parciales). Toda comunicación de datos necesita su tiempo, por lo que cuando se diseñan y usan sistemas Big Data hay que tener ser consciente de la diferencia en tiempo según dónde se ubiquen los procesos que se van a comunicar.

Comunicación dentro de la misma máquina:

Es el caso de comunicación más rápido posible, ya que los datos no necesitan ser enviados por red sino que los procesos pueden intercambiarlos a través recursos residentes en la propia máquina.

- Memoria RAM.
- Sistema de ficheros.
- Base de datos (útil pero lento).

Comunicación entre dos máquinas dentro del mismo switch:

La comunicación entre máquinas a través de una red de comunicaciones siempre es varios órdenes de magnitud más lenta que si se realiza entre procesos residentes en la misma máquina. Esto se debe a que es necesario enviarla al interfaz de red, este tiene que enviarla utilizando el protocolo de comunicaciones que corresponda, y la otra máquina tiene que leer la información recibida desde su propio interfaz de red.

Hay que tener en cuenta que, si bien es más lento, este tipo de comuncación es totalmente obligatoria en entornos de Big Data (los cuales precisamente se basan en el uso de distintos nodos dentro de un cluster).

Llegados al punto de necesitar comunicar datos entre máquinas, el modo más rápido es que sean nodos conectados al mismo switch, ya que es el caso en el que la información necesita realizar el menor número de saltos entre elementos de la red.

Comuncación fuera del switch:

Cuantos más saltos entre switch interconectados necesite dar la información para llegar de un nodo a otro, más lenta será la comunicación.

El número de bocas de los switchs es limitado, por lo que típicamente todos los servidores dentro de un mismo rack en un CPD están conectados a un mismo switch, y hay una jerarquía de switches mediante la cual se comunica a unos racks con otros. El software que gestiona la infraestructura del clúster se configura con información acerca de la jerarquía que se ha utilizado a la hora de realizar esas conexiones físicas entre los nodos, de modo que pueda tomar las mejores decisiones a la hora de emplazar tanto datos como procesado en los nodos, con el fin de minimizar el número de veces que la información tiene que saltar entre switches.

Comunicación fuera del CPD:

Los nodos del cluster estarán típicamente todos dentro del mismo CPD, pero los datos no se originan dentro del CPD sino fuera de él, por lo que habrá que hacerlos llegar al mismo. De igual modo, los resultados producidos por el cluster en gran cantidad de casos son para ser consumidos o utilizados fuera de su CPD, por lo que habrá que sacarlos del él.

Esta comunicación se realiza a una velocidad muy inferior a la que se puede obtener a través de un switch, de modo que este es el tipo de comunicación más lento de todos.

Autoevaluación

A la hora de enviar datos entre dos nodos de un clúster, ¿en qué caso será más rapida la comunicación?

- ○ Es vital que haya pocos metros de cable entre los nodos.
- ○ Será más rápido cuantos menos saltos entre switches haya que hacer para llegar de un nodo a otro.
- ○ Lo más rápido siempre es enviarlos a otro CPD si éste cuenta con máquinas más rápidas.
- ○ Los nodos de un clúster no pueden comunicarse entre sí.

Incorrecto. La longitud de los cables es algo que marque la diferencia.

Correcto.

Incorrecto. Enviar datos entre CPDs es mucho más lento que comunicarse dentro de uno de ellos.

Incorrecto. La clave es precisamente que pueden comunicarse.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto
4. Incorrecto

4.3.- Estrategias de procesamiento de datos.

En entornos Big Data se emplean distintas estrategias a la hora de procesar los datos, las cuales se escogen según la cantidad y naturaleza de los mismos, así como de las necesidades de la actividad que se esté realizando.

- Por lotes (del inglés batch).
- Transaccional.
- En tiempo real (del inglés realtime).
- Streaming (anglicismo usado sin traducir al castellano).

Por lotes:

El procesamiento por lotes (o también en inglés offline, por contraposición a online, que denota el de tiempo real), se realiza sin la necesidad de producir respuestas en un corto plazo. Pueden tardar en ejecutarse horas o incluso días.

Esta estrategia se emplea mayormente para trabajo de analítica con gran cantidad de datos (en ocasiones todos los disponibles para la tarea en cuestión).

Transaccional:

Al contrario del caso del procesamiento por lotes en el que el tiempo transcurrido hasta que se produce una respuesta no es demasiado importante, en el caso de las tareas transaccionales es de obligado cumplimiento que el tiempo necesario sea muy corto (a ser posible siempre por debajo de un segundo).

Este tipo de procesamiento se emplea cuando se realizan transacciones. Debido a restricciones en cuanto a tiempo de las tareas transaccionales, este tipo de procesamiento no puede afectar a un gran volumen de datos.

En tiempo real:

Al igual que en el caso del procesamiento transaccional, este tipo de procesamiento produce resultados en un corto espacio de tiempo.

Se emplea para analíticas interactivas (por lo general de tipo descriptivo), en las que un usuario humano está consultando estadísticas acerca de los datos (razón para que el tiempo de respuesta deba ser pequeño).

Para poder realizarse con un gran volumen de datos se suelen emplear subsistemas de tipo OLAP, en muchas ocasiones almacenadas en

memoria.

Es importante destacar que en muchas ocasiones escucharemos decir que el procesamiento transaccional se produce en tiempo real, lo cual es totalmente cierto. En otras palabras, "tiempo real" puede en la práctica emplearse tanto para procesamiento analítico como para actividades transaccionales.

Al contrario no ocurre lo mismo. Es decir, no sería correcto decir que un procesamiento analítico en tiempo real es transaccional, ya que el término transaccional ya incluye la existencia de una transacción, lo cual no está ocurriendo cuando lo que estamos haciendo es analizar datos.

Streaming:

El procesamiento en streaming tiene mucho que ver con el que se produce en tiempo real en cuando a que debe tener un corto tiempo de respuesta. Sin embargo, en este caso la clave es que ha de producirse a la velocidad a la que se recibe el flujo (de ahí streaming) de datos de entrada.

Esto añade una complejidad extra a los sistemas que han de diseñarse para ser capaces de procesar/analizar datos en streaming. Esto se debe a que las estructuras de datos en las que se mantiene la información necesaria para realizar la analítica deben ser capaces de actualizarse a la medida que llegan nuevos datos. Por ello, necesitan almacenar esa información en memoria, lo que implica un tope máximo en el tamaño de datos que pueden tratarse a la vez.

Autoevaluación

¿Cuál es la diferencia entre procesamiento en tiempo real y procesamiento en streaming?

- ○ Tiempo real implica que los resultados se producen en poco tiempo, mientras que en streaming implica que es capaz de tener en cuenta datos que van entrando constantemente.
- ○ Son lo mismo.
- ○ Streaming significa que es transaccional, mientras que en tiempo real significa que no es transaccional.
- ○ Streaming significa que el procesamiento es rápido, mientras que en tiempo real significa que todo ocurre a la velocidad a la que lo solicita el usuario.

Correcto.

Incorrecto. Cada cual tiene su propio significaco.

Incorrecto. Streaming no tiene nada que ver con transaccional, y tiempo real muchas veces es transaccional.

Incorrecto. El significado es otro en ambos casos.

Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto
4. Incorrecto

4.4.- OLTP.

Empleamos el acrónimo OLTP para designar un sistema que está orientado a transacciones, por lo que trata con los datos operacionales del día a día, relacionados con acciones que necesitan realizarse en tiempo real (de ahí que se llamen "online").

Tales transacciones son por lo general realizadas contra bases de datos relacionales, incluyendo sólo acciones sencillas (insertar, seleccionar, actualizar o eliminar) sin ningún tipo de analítica, gracias a lo cual se consiguen tiempos de respuesta inferiores a un segundo de modo que el sistema sea usable sin producir tiempos de espera desagradables para los usuarios.

Para saber más

En este enlace puedes ver más información acerca de lo que significa OLTP.

OLTP

Autoevaluación

Cuando hablamos de OLTP, ¿qué tipo de base de datos se está empleando por lo general?

- ○ Una base de datos orientada a grafos.
- ○ No será una base de datos común sino que todo se estará almacenando en la memoria RAM del sistema.
- ○ Una base de datos relacional.
- ○ No se emplea una base de datos sino un almacenamiento distribuído como HDFS o S3.

Incorrecto. Una base de datos orientada a grafos puede llegar a ser transaccional, pero no es elección común.

Incorrecto. Por lo general son bases datos de las consideradas "comunes".

Correcto.

Incorrecto. Los almacenamientos distribuidos de esos tipos no son para OLTP.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

4.5.- OLAP.

Empleamos el acrónimo OLAP para designar un sistema que está orientado procesar consultas de tipo analítico en tiempo real. Este tipo de sistemas son parte integral de la inteligencia de negocio (BI) y la minería de datos, usándose en todos los niveles de la analítica de datos (desde el análisis descriptivo hasta el prescriptivo).

Almacenan los datos en bases de datos multidimensionales (en ocasiones llamadas "cubos OLAP" debido a dicha estructura multidimensional), altamente optimizadas para poder responder en muy poco tiempo a consultas complejas que afectan a lo que en el mundo relacional/transaccional correspondería a varias tablas. Para obtener tal rendimiento, tales bases de datos multidimensionales guardan los datos denormalizados. En ocasiones se mantienen en la memoria RAM de la máquina que ejecuta el sistema, lo cual en tal caso implica un límite máximo respecto de la cantidad de datos que se pueden tratar a la vez.

Debes conocer

Para entender qué significa que en OLAP se guarden los datos denormalizados, antes necesitas entender qué es la normalización de datos que emplea en el mundo de las bases de datos relacionales.

Accede al siguiente enlace para ver más información:

Normalización de bases de datos

Para saber más

En el siguiente enlace puedes ver más información sobre lo que significa OLAP.

OLAP

Autoevaluación

¿Cuáles son rasgos típicos de las estructuras de datos empleadas para OLAP?

- Almacenamiento en unidades SSD para un acceso más rápido.
- Almacenamiento en memoria RAM de estructuras multidimensionales.
- Almacenamiento en memoria RAM de datos previamente normalizados.
- Estructuras multidimensionales que se almacenan en sistemas distribuídos tipo HDFS o S3.

Incorrecto. Parte del sistema puede emplear SSDs, pero no es una característica distintiva.

Correcto.

Incorrecto. Los datos normalizados los encontramos en bases de datos relacionales.

Incorrecto. Los datos pueden provenir de ese tipo de sistemas distribuido, pero las estructuras multidimensionales a emplear para OLAP por lo general no están en ellos sino en memoria.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto
4. Incorrecto

4.6.- Principio SCV.

Mientras que el teorema CAP tiene que ver con almacenamiento de datos distribuídos, el principio SCV está relacionado con el procesamiento distribuído de los datos. Es decir, no tiene que ver con la escritura y lectura (consistente o no) de los datos en entornos distribuídos sino con el procesamiento que se realiza sobre ellos dentro de los nodos de un sistema de procesamiento distribuído.

De modo similar a lo que ocurría con el teorema CAP, el principio SCV establece que un sistema de procesamiento distribuído sólo puede soportar como máximo 2 de las siguientes 3 características.

- Velocidad (**S**peed).
- Consistencia (**C**onsistency).
- Volumen (**V**olume).

Velocidad:

Se refiere a cuánto tardan en procesarse los datos desde el momento en el que son recibidos en el sistema analítico. Por lo general se excluye el tiempo que se tarda en capturar los datos, considerando sólo lo que se tarda en generar la estadística o ejecutar el algoritmo en cuestión.

Esta velocidad es más alta si estamos ante un sistema de analítica en tiempo real que si se trata de un sistema de analítica por lotes (del inglés batch).

Consistencia:

Se refiere en este caso a la precisión de los resultados de la analítica (no confundir, por lo tanto, con el significado de la C del teorema CAP).

Tal precisión depende de si para la analítica se utilizan todos los datos disponibles (precisión alta) o de si por el contrario se emplean técnicas de muestreo para seleccionar sólo un subconjunto de los mismos con la intención de producir resultados (de menor precisión) en un menor tiempo.

Volumen:

Se refiere a la cantidad de datos que pueden ser procesados.

Hay que tener en cuenta que en entornos de Big Data, el alto volumen de datos es una características siempre presente (una de las 5 Vs).

De igual modo que hicimos al estudiar el teorema CAP, nos fijaremos en una serie de escenarios para mostrar que no podemos conseguir un sistema que cumpla a la vez las 3 características del principio SCV.

- Si se requiere velocidad (S) y consistencia (C), no podemos procesar un alto volumen (V) de datos ya que eso aumenta el tiempo de respuesta.
- Si se requiere consistencia (C) y poder procesar grande volúmenes de datos (V), no es posible realizar tal procesado a una alta velocidad (S).
- Si necesitamos procesar un alto volumen de datos (V) a una alta velocidad (S), entonces necesitaremos emplear técnicas de muestreo para seleccionar sólo un subconjunto de esos datos, lo cual producirá un resultado no consistente (C).

Reflexiona

Dado que en ambientes de Big Data el ser capaz de manejar grandes volúmenes de datos (V) es una obligación, ¿podremos típicamente realizar analítica en tiempo real con todos ellos?

Mostrar retroalimentación

No, ya que para que sea en tiempo real debemos cumplir S, por lo que necesitaríamos realizar la analítica sobre un subconjunto de los datos, encontrándonos en el caso S+V, y produciendo por lo tanto un resultado no totalmente consistente (C).

Reflexiona

Dado que en ambientes de Big Data el ser capaz de manejar grandes volúmenes de datos (V) es una obligación, ¿podremos típicamente realizar procesamiento por lotes empleando todo ese conjunto de datos?

Mostrar retroalimentación

Sí, ya que al ser analítica por lotes en lugar de en tiempo real, la característica S ya no es necesaria, de modo que podemos encontrarnos en un caso C+V, utilizando todos los datos sin ningún tipo de muestreo para así producir un resultado consistente (C).

5.- La arquitectura por capas de Big Data.

Caso práctico

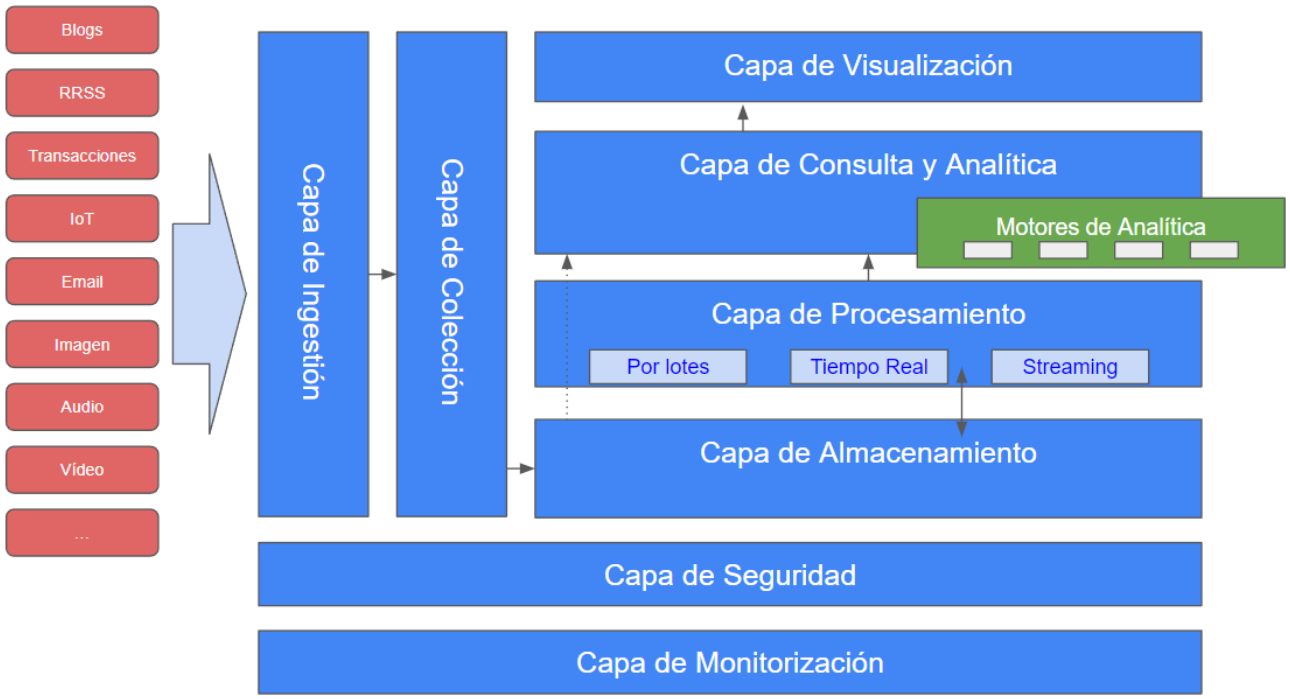
Estamos en el año 2002. **Susana** y **Silvia** están sentadas frente a un motón de papeles. La empresa para la que trabajan les ha encargado un nuevo sistema que permita tratar grandes cantidades de datos.

Christina Morillo (Dominio público)

—Menudo lío estamos montando —dice Susana—. Este programa tiene que hacer demasiadas cosas distintas, ¿no te parece?

—Oye, se me está ocurriendo una cosa —dice Silvia—. ¿Y si en lugar de un programa que haga todo esto hacemos distintos programas, uno para cada cosa, y luego hacemos que se comuniquen?

Capas de Big Data



Víctor Tomico (Dominio público)

Al margen de que durante el diseño y desarrollo de cada posible proyecto de Big Data pueda optarse por la estructura o arquitectura que más convenga, de modo generalizado se emplea una arquitectura según la cual el flujo de datos va pasando por una serie de

capas.

- **Capa de ingestión:**
  Como primer paso, los datos se obtienen desde múltiples fuentes con las cuales es necesario conectarse de algún modo. Hay que tener en cuenta que la gran mayoría de las fuentes son preexistentes a la creación del sistema Big Data que se esté desarrollando, por lo que es el sistema el que tiene que adaptarse a las fuentes y no a la inversa (empleando el procolo correspondiente a las mismas y siendo capaz de interpetar los datos que de ellas se obtienen).

- **Capa de colección:**
  Una vez que se obtienen e interpretan los datos viene el trabajo relacionado con integrarlos para darles una estructura propia. Hay que tener en cuenta que las fuentes de datos pueden ser muchas y de naturaleza muy variada, cada una emitiendo información en un formato distinto, de modo que hay que unificarlo todo para representarlo como un único conjunto de datos con sentido y ya prácticamente listos para ser utilizados.

- **Capa de almacenamiento:**
  Como no podía ser de otro modo, esa gran cantidad de datos que da origen al concepto Big Data debe ser almacenada, para lo cual se emplean sistemas de almacenamiento distribuido especialmente diseñados para ello.

- **Capa de procesamiento:**
  Es la capa que provee de infraestructura a la siguiente capa (la de consulta y analítica) para poder tratar con gran cantidad de datos. Es decir, facilita el procesamiento (por lotes, en tiempo real, streaming o híbrido) pero únicamente hace lo que le está pidiendo la capa superior, no obteniendo valor del dato de por sí.

- **Capa de consulta y analítica:**
  Es la capa en la que se comienza a obtener valor al dato, realizando la estadística, algoritmia o análisis que se considere oportuno, para ello siempre basándose en la capa previa de procesamiento.

- **Capa de visualización:**
  Es la capa con la que interacciona el usuario final, el cual puede consultar reportes estáticos o acceder a cuadros de mando interactivos con diversas visualizaciones y controles desde los cuales puede decidir qué información ver y cómo quiere verla representada. Desde esta capa es desde por lo general se toman las decisiones de negocio.

- **Capa de seguridad:**
  Capa transversal que da soporte a todo lo relacionado con asegurar la seguridad en los datos empleando métodos tanto físicos como de software. Incluye protección ante el ataque o uso malintencionado tanto desde dentro como desde fuera de la empresa o institución.

- **Capa de monitorización:**
  Capa transversal que da soporte a todo lo relacionado con la monitorización tanto de los datos como del propio sistema. La monitorización de datos incluye auditoría, testeo, gestión y control, de modo que los datos a emplear para obtener valor sean correctos y frescos. Tal monitorización es una parte importante de los mecanismos de gobernanza de datos.

Para saber más

Puedes ver más sobre lo que significa seguridad de la información en el siguiente enlace:

Seguridad de la información

Para saber más

Puedes ver más información sobre lo que significa gobernanza de datos en el siguiente enlace:

Gobernanza de datos

Autoevaluación

¿Qué capa se encarga de integrar los datos de modo que queden unificados con sentido propio para la tarea que se va a realizar con ellos?

- La capa de ingestión.
- La capa de colección.
- La capa de almacenamiento.
- La capa de procesamiento.

Incorrecto. Esa es la capa de entrada de los datos.

Correcto.

Incorrecto. Esa es la capa en la que se guardan los datos.

Incorrecto. Esa es la capa en la que se procesan los datos.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto
4. Incorrecto

6.- El paisaje de Big Data.

Caso práctico

La conversación de cafetería de todos los viernes a última hora se está poniendo bastante más animada que de costumbre.

—Entonces lo que vamos a usar para adquirirlos es Hadoop, ¿no? —dice **Irene**.

—Bueno, no exactamente. Hadoop es la plataforma, pero para adquirir los datos desde las fuentes usaremos **Flume** o **Sqoop** según el caso —dice **Marcos**.

—Vale, ¿y entonces estos que hay aquí apuntados? ¿**Pig** y **Hive**?

—Esos los usaremos para hacer consultas una vez estén los datos cargados en **HDFS**.

—Y con eso es con lo que analizamos los datos, ¿verdad?

Eduardo Simões Neto Junior (Dominio público)

—Bueno, algo se puede hacer, pero el verdadero análisis lo haremos mediante **Spark**, y lo que es aprendizaje automático lo haremos con **H_2O**.

—¿Y con **Spark** ya podemos sacar las visualizaciones?

—Eso lo haremos o con **Power BI** o con **Tableau**.

Hablamos de paisaje Big Data (más usado en inglés, como "The Big Data Landscape") para referirnos al panorama de las diversas herramientas y utilidades que se pueden emplear para desarrollar proyectos Big Data, muchas veces categorizadas según la capa de procesamiento a la que pertenecen o según el tipo de actividad que realizan.

Desde los inicios de Big Data, diversos autores han ido creando collages (unos más detallados que otros), tratando de capturar la riqueza de tal panorama.

Por esta razón, en esta sección no vamos a incluir una imagen en concreto sino que vamos a sugerir al alumno que realice la búsqueda "big data landscape" en su buscador preferido.

Debes conocer

Accede a tu buscador favorito y realiza la búsqueda "big data landscape" para encontrar gran multitud de imágenes que te muestran diversas interpretaciones de lo que es el paisaje (o panorama) de Big Data.

En el siguiente artículo también puede ver una introducción al ecosistema Hadoop y una imagen resumen:

El Ecosistema Hadoop (III) : Una gran diversidad “biológica"

En esas imágenes podrás encontrar, entre otras cosas:

- **Hadoop**, como la plataforma pionera para Big Data, enfocada a trabajo por lotes.
- Una gran **variedad de herramientas** pertenecientes al ecosistema de *Hadoop*, diseñada cada una de ellas para una función dentro de las distintas fases que componen el trabajo con datos. Entre otras:
  - **HDFS**, es la capa de almacenamiento de *Hadoop*, y como tal, es un sistema de ficheros distribuido y tolerante a fallos que puede almacenar gran cantidad de datos, escalar de forma incremental y sobrevivir a fallos de hardware sin perder datos.
  - **Sqoop**, herramienta diseñada para transferir de forma eficiente datos crudos entre un clúster de *Hadoop* y un almacenamiento estructurado, como una base de datos relacional.
  - **Flume**, software para tratamiento e ingesta de datos masivo, facilitando crear desarrollos complejos que permiten el tratamiento de datos en streaming.
  - **Hive**, es una tecnología distribuida diseñada y construida sobre un clúster de *Hadoop*. Permite leer, escribir y gestionar grandes datasets (con escala de petabytes) que residen en HDFS haciendo uso de un lenguaje dialecto de SQL, conocido como *HiveSQL*, lo que simplifica mucho el desarrollo y la gestión de *Hadoop*.
- **Spark**, como plataforma enfocada a procesamiento en tiempo real y/o streaming, capaz de interactuar con muchas de las herramientas ya disponibles en el ecosistema Hadoop (de hecho según el punto de vista, *Spark* podría considerarse una herramienta más dentro del ecosistema).
- **Nifi**, es un proyecto de *Apache* (desarrollado en Java inicialmente por la NSA), que plantea un sistema distribuido dedicado a ingestar y transformar datos mediante un modelo en streaming.
- **Kafka**, es un middleware de mensajería entre sistemas heterogéneos, que mediante un sistema de colas facilita la comunicación asíncrona, desacoplando los flujos de datos de los sistemas que los producen o consumen. Funciona como un broker de mensajes, encargado de enrutar los mensajes entre los clientes de un modo muy rápido.
- Bases de datos **NoSQL** y **NewSQL** como soluciones de almacenamiento para necesidades y casos específicos.
- Diversas **herramientas para analítica**, las cuales se pueden a su vez subclasificar.
- Diversas **herramientas para visualización**.
- Diversas **aplicaciones específicas** se nutren de o interaccionan con alguna de las

herramientas ya comentadas.

Si nos centramos en qué **uso** se le da hoy en día al Big Data, por citar algunos, sería:

- La industria 4.0 se basa en gran medida en almacenar grandes cantidades de datos provenientes de sensores IoT.
- Las operadores de streaming con sus motores de recomendaciones y trazabilidad de la navegación del usuario se basan en guardar todos los eventos generados al navegar y utilizar sus servicios.
- Las aplicaciones de mapas que nos recomiendan diferentes rutas en base al tráfico actual lo hacen a partir de los datos de todos los conductores, tanto del histórico como de los presentes.
- Las redes sociales son, en parte, la cuna del Big Data y fuente de los últimos modelos generativos que requieren de mucha información para su entrenamiento. Además, como fuente de publicidad, es muy común realizar un análisis de sentimiento de los mensajes de los usuarios a modo de termómetro del lanzamiento de un producto o sondeo de la población.

Autoevaluación

¿Qué queda representado en el paisaje de Big Data?

- ○ Las distintas capas por las que pasan los datos.
- ○ La posible distribución de los nodos de un clúster dentro de un centro de datos.
- ○ Las herramientas y utilidades que se pueden utilizar.
- ○ Las herramientas y utilidades que sirven para obtener datos de diversas fuentes.

Incorrecto. Exite una arquitectura por capas, pero el paisaje es otra cosa.

Incorrecto. Eso sería en todo caso un mapa de nodos, pero no tiene nada que ver con el paisaje de Big Data.

Correcto.

Incorrecto. No se limita a ese tipo de herramientas y utilidades.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

6.1.- Roles y empleos.

Hay una amplia variedad de roles implicados en la administración, el control y el uso de datos. Algunos roles están orientados a los negocios, mientras que otros implican más ingeniería. También los hay más centrados en la investigación, o incluso existen roles híbridos que combinan distintos aspectos de la administración de datos. La organización puede definir roles de maneras distintas o asignarles nombres diferentes, pero los que se describen a continuación:

Administrador de base de datos

Un administrador de base de datos es responsable del diseño, la implementación, el mantenimiento y los aspectos operativos de los sistemas de bases de datos locales y los basados en la nube. Son responsables de la disponibilidad general y de las optimizaciones y el rendimiento coherentes de las bases de datos. Trabajan con las partes interesadas para implementar directivas, herramientas y procesos para la realización de copias de seguridad, así como planes de recuperación que permiten reponerse tras un desastre natural o un error humano.

Los administradores de base de datos también son responsables de administrar la seguridad de los datos en la base de datos, conceder privilegios sobre los datos, y conceder o denegar el acceso a los usuarios según corresponda.

Un administrador de base de datos no es un rol o empleo exclusivo de Big Data, pero desempeña un papel importante en el proceso.

Ingeniero de datos

Los ingenieros de datos colaboran con las partes interesadas para diseñar e implementar cargas de trabajo relacionadas con datos, incluidas canalizaciones de ingesta de datos, actividades de limpieza y transformación, y almacenes de datos para cargas de trabajo analíticas. Usan una amplia gama de tecnologías de plataforma de datos, como bases de datos relacionales y no relacionales, almacenes de archivos y flujos de datos.

También son responsables de garantizar que la privacidad de los datos se mantenga dentro de la nube y que abarque desde el entorno local hasta los almacenes de datos en la nube. Se ocupan de la administración y la supervisión de canalizaciones de datos para asegurarse de que las cargas de datos funcionen según lo previsto.

En resumen, los ingenieros de datos crean y operan la infraestructura de datos necesaria para preparar los datos para su posterior análisis por parte de analistas de datos y científicos.

Analista de datos

Los analistas de datos ayudan a las empresas a maximizar el valor de sus recursos de datos. Son los responsables de explorar datos para identificar tendencias y relaciones, diseñar e implementar modelos analíticos, y habilitar funcionalidades de análisis avanzado mediante informes y visualizaciones.

Los analistas de datos se ocupan del procesamiento de los datos sin procesar para convertirlos en información pertinente, en función de los requisitos empresariales establecidos, con el fin de ofrecer conclusiones de interés.

Los analistas de datos consultan, procesan, proporcionan informes y resumen y visualizan datos. Aprovechan las herramientas y los métodos existentes para resolver un problema. Ayudan a las personas, como los analistas de negocio, a comprender consultas específicas con informes y gráficos ad hoc. Los analistas de datos deben comprender los principios estadísticos básicos, la limpieza de diferentes tipos de datos, la visualización y el análisis exploratorio de datos.

En resumen, los analistas de datos analizan los datos para ayudar a las empresas y otras organizaciones a tomar decisiones informadas.

Científico de datos

Los científicos de datos aplican las estadísticas, el aprendizaje automático y los enfoques analíticos para responder las preguntas esenciales de la empresa. Interpretan y entregan los resultados de sus hallazgos mediante el uso de técnicas de visualización, la creación de aplicaciones de ciencia de datos o la narración de historias emocionantes sobre las soluciones a sus problemas de datos (empresariales).

Trabajan con los conjuntos de datos de diferentes tamaños, y ejecutan algoritmos en los grandes conjuntos de datos. Los científicos de datos deben estar al día con las últimas tecnologías de automatización y aprendizaje automático.

Los requisitos para desempeñar estos roles incluyen habilidades estadísticas y analíticas, conocimientos de programación (Python, R, Java) y familiaridad con Hadoop, un conjunto de utilidades de software de código abierto que facilita el trabajo con grandes cantidades de datos.

En resumen, los científicos de datos son expertos en datos que organizan y ofrecen valor a partir de los datos.

Reflexiona

Existen diferencias clave entre los roles de Analista de datos y Científico de datos, pero comparten el mismo objetivo: traducir el análisis de datos en

inteligencia empresarial.

En la siguiente tabla podemos ver una serie de características de cada uno de los roles y de ambos.

Analista de datos	Ambos	Científico de datos
✓ Recopilación de datos. ✓ Minería de datos. ✓ Almacenamiento de datos. ✓ Limpieza de datos. ✓ Visualización de datos. ✓ Uso de herramientas como Microsoft Excel, Tableau, Power BI,...	✓ Pensamiento analítico. ✓ Pensamiento crítico. ✓ Narración de datos. ✓ Habilidades interpersonales. ✓ Uso de lenguajes de programación como Python.	✓ Análisis estadístico. ✓ Análisis predictivo. ✓ Aprendizaje automático. ✓ Entrenamiento e implementación de modelos. ✓ Uso de herramientas como Apache Hadoop.

Fuente: Data Analytics vs. Data Science by CompTIA.