CURSO:
BASES DE DATOS
Unidad I: Modelamiento de datos.

Clase 2.1: Distribución de los datos.

Profesor: Diego Miranda

Data Scientist

INTRODUCCIÓN

Consideremos de nuevo el ejemplo de WebShop. ¿Qué pasa si WebShop crece mucho como empresa, y se encuentra con 100.000.000 usuarios, y 1.000.000.000.000 de transacciones por día? (como una observación: las empresas como Amazon o Google tienen más usuarios que esto hoy en día) ¿Creen que estos datos se podrían guardar como antes en un servidor corriendo una base de datos común y corriente? Como la cantidad de los datos que WebShop debería guardar en este caso cada día es probablemente más grande que la capacidad de un disco duro moderno, la empresa obviamente tiene que usar otro modo de manejo para sus datos. Cuando esto pasa, podemos considerar que la empresa necesita hacer el paso al mundo de Big Data.

PASO AL MUNDO DEL BIG DATA

Las empresas como el WebShop tienen todos sus datos en un computador, y usan un motor de bases de datos para procesar todas las tareas en este computador. Típicamente este computador es un servidor, y permite a varios usuarios conectarse y hacer ciertas tareas que utilizan la base de datos. Por ejemplo, cuando un cliente hace su compra, el software del WebShop se conecta al servidor, y registra la compra en la base de datos. Similarmente, cuando el equipo de despachos registra un despacho, igual tienen que cambiar los datos en la base de datos que reside en el servidor.

BASE DE DATOS CENTRALIZADA

Este tipo de sistemas donde todo ocurre en el mismo computador/servidor se llaman sistemas centralizados, porque todas las tareas ocurren en un lugar (el servidor en nuestro caso), y típicamente utilizan un procesador conectado con la memoria para guardar los datos. Los sistemas de manejo de datos centralizados son los más usados en la práctica, y ofrecen una gran cantidad de ventajas comparado con otros tipos de sistemas.

VENTAJAS

- Es fácil maximizar la integridad de los datos, porque el sistema tiene el control sobre todos los datos al mismo tiempo.
- La redundancia de los datos es minimizada, porque es suficiente guardar solo una copia de los datos (en caso de qué uno quisiera asegurarse contra fallas, podría ser necesario guardar copias).
- Son más fáciles para administrar y manejar.
- Generalmente más baratos que otros tipos de sistemas, porque requieren una máquina, y una licencia de software para el manejo de los datos.
- Pueden ser más seguros porque existe solo un punto de ataque (el servidor central).

LIMITACIONES

Dadas las buenas características de los sistemas de manejo de datos centralizados y dado que este tipo de sistemas es perfectamente adecuado para la gran mayoría de las empresas pequeñas, no es sorprendente que sistemas centralizados son usados muy frecuentemente en la práctica. El problema es cuando la cantidad de los datos que se necesitan guardar crece tanto que los datos ya no caben en el disco duro del servidor de la empresa. En este caso tenemos 2 opciones:

1. Comprar un servidor más potente que puede guardar todos los datos que necesitamos.
2. 2. Comprar otro servidor de la misma potencia y utilizar los dos al mismo tiempos.

SOLUCIÓN 1

Obviamente, la solución 1 arriba parece más sencilla, porque nos permite replicar el mismo modelo en el nuevo servidor, pero ¿qué pasará cuando los datos vuelvan a crecer? ¿Qué pasará cuando los datos crecen n veces? Quizás el nuevo servidor pueda manejar la primera o incluso la segunda duplicación en el tamaño de los datos, pero el problema con esta solución es que en el largo plazo no escala: quiere decir, el crecimiento del volumen de los datos es generalmente mucho más rápido que el crecimiento en poder de las máquinas.

SOLUCIÓN 2

Por este motivo, para manejar grandes volúmenes de datos, la técnica más usada hoy en día es utilizar varias máquinas (muchas veces idénticas), que guardan una parte de los datos (o la copia de todos los datos), e intentan resolver uno o más problemas de manera concurrente. Como los datos quedan distribuidos en distintas máquinas, los sistemas para el manejo de los datos de esta manera se llaman sistemas distribuidos de manejo de los datos.

SISTEMAS DISTRIBUIDOS

Distribución no se usa solo para manejo de los datos, sino también en cualquier sistema que no necesita resolver todas las tareas en un lugar central. Ejemplos de estos tipos de sistemas son: la Internet y la Web, el control del tráfico aéreo, servicios web grandes como Google, Amazon, o Facebook. Formalmente, un sistema distribuido es un sistema que permite a una colección de computadores independientes comunicarse a través de una red, para resolver un problema común.

CARACTERÍSTICAS SISTEMAS DISTRIBUIDOS

- 1. Concurrencia. Quiere decir que los computadores involucrados en un sistema distribuido trabajan al mismo tiempo, sin afectar el funcionamiento de otros computadores en el sistema.

- 2. Fallas independientes (eng. “independent failures”). Quiere decir que una máquina en el sistema puede fallar sin afectar el funcionamiento del sistema.

- 3. No hay un reloj global. Quiere decir que las máquinas en un sistema distribuido no agendan sus operaciones según un reloj compartido, sino que se comuniquen una con la otra a través de la red cuando necesitan intercambiar información.

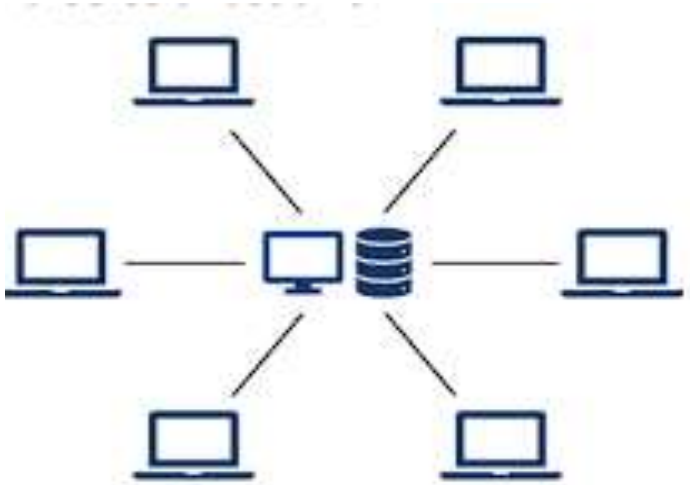
PROPIEDADES SISTEMAS DISTRIBUIDOS

- **Transparencia:** Aunque un sistema distribuido involucra muchas máquinas, se busca que se vean como un sistema único. Por ejemplo, si uno accede su cuenta de correo de Google, no debería preocuparse si se conectó a un servidor en Chile, o a un servidor en Estados Unidos. Esto normalmente se logra usando una interfaz abstracta para acceder a las máquinas y datos de un sistema distribuido.
- **Flexibilidad:** El sistema debería permitir agregar máquinas nuevas y desconectar las existentes con facilidad. Esto normalmente se logra usando la replicación de los datos y de las tareas.
- **Confiabilidad (eng. "Reliability")**: Uno también quiere que el sistema funciona bien, y será resistente. Esto quiere decir que el sistema pueda evadir las fallas, o que pueda seguir funcionando cuando hay fallas en algunas máquinas del sistema.
- **Eficiencia:** Es sistema es eficiente, y puede resolver tareas rápido, y utilizando pocos recursos. Quiere decir que el usuario final no necesita esperar a su respuesta por mucho tiempo.
- **Escalabilidad:** Según las necesidades, el sistema debe poder crecer fácilmente agregando más máquinas. Esto permite al sistema procesar una cantidad más grande de datos o hacer tareas más complejas.

ARQUITECTURA SISTEMAS DISTRIBUIDOS

ARQUITECTURA SERVIDOR CLIENTE

En este caso, existe uno o más servidores que guardan los datos y hacen tareas intensivas de cómputo. Por otro lado, también existe uno o más clientes que se preocupan de tareas sencillas como la interfaz gráfica para los usuarios, o como permitir a los usuarios ejecutar sus tareas. Un ejemplo de esta arquitectura son los servicios web de un banco, donde la página web o aplicación móvil actúan como clientes y un usuario puede pedir la información sobre sus cuentas. Para recuperar estos datos, la página web (actuando como un cliente) se conecta al servidor de banco que tiene un DBMS con los datos y ejecuta la consulta necesaria para recuperar los datos deseados en el servidor. Finalmente, el servidor devuelve estos datos a la página web para mostrárselos al usuario.

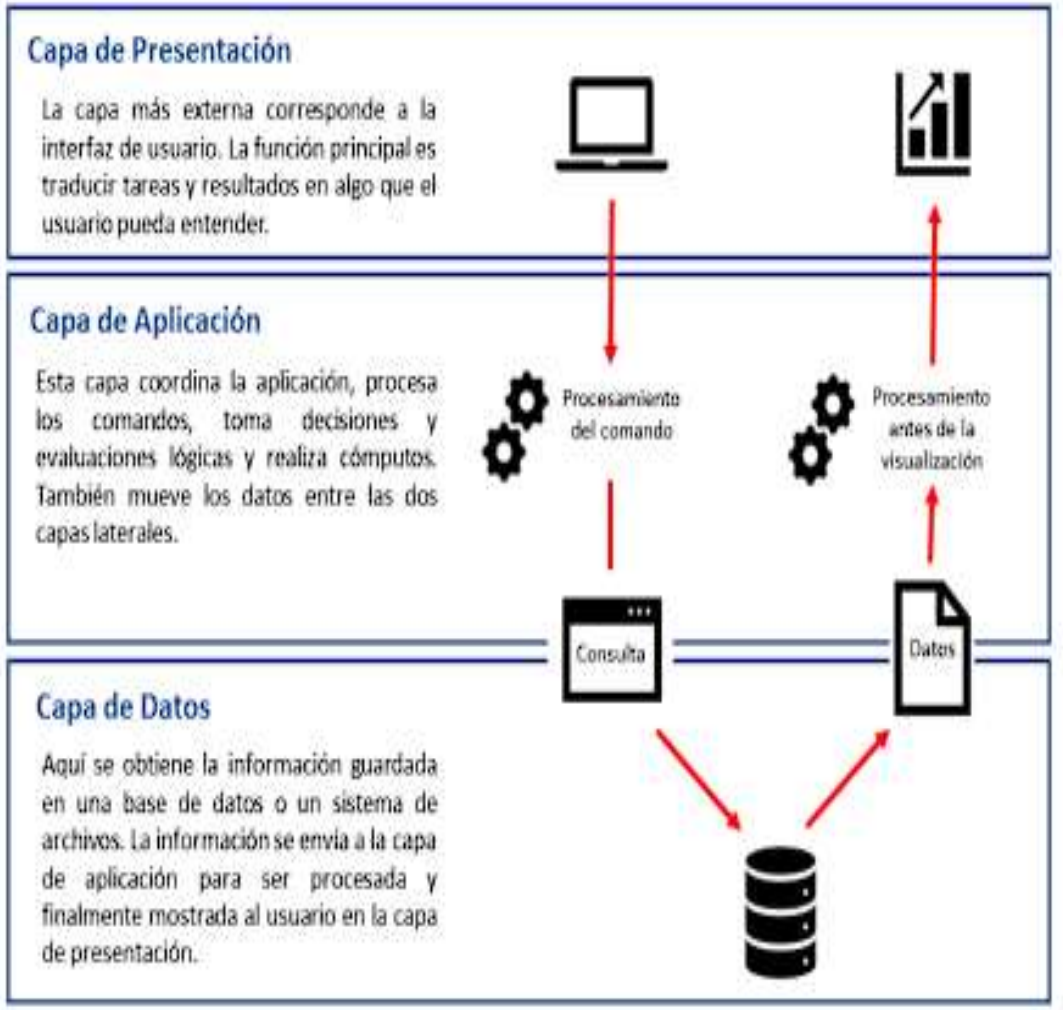


ARQUITECTURA DE TRES CAPAS

- **a. Capa de presentación**, qué se preocupa solo de la interfaz de usuario, facilita el acceso a los datos, permite hacer consultas de manera sencilla, y despliega los resultados. En la arquitectura servidor-cliente anterior esto era realizado por la página web.

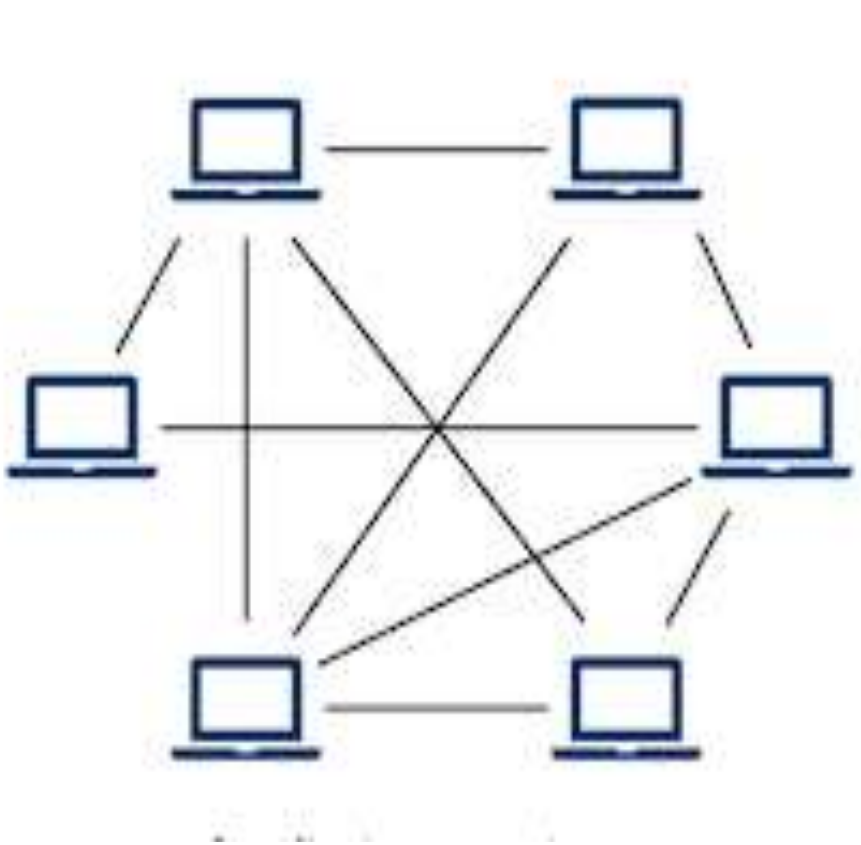
- **b. Capa de aplicación**, que se preocupa de procesar los datos y consultas. Capa de aplicación recibe lo que el usuario quiere de la capa de presentación, lo transforma a una consulta, y la pasa a capa de datos. Cuando la capa de datos devuelve la información, la capa de aplicación procesa a los datos para obtener la información para el cliente, y la pasa a la capa de presentación para entregársela al cliente.

- **c. Capa de datos**, que contiene las máquinas corriendo sistemas de manejo de los datos, y permite a la capa de aplicación acceder a estos datos. La idea de la arquitectura de tres capase es separar los procesos de un sistema distribuido tal que cada capa funciona de manera independiente.



ARQUITECTURA DE PARES

En esta arquitectura todas las máquinas tienen el mismo rol en la red y no interactúan basadas en una coordinación central, sino que se comunican una con la otra directamente. En cierto sentido, en una red de pares cada máquina actúa tanto de servidor como de cliente. Ejemplos de esta arquitectura son bittorrent, o criptomonedas como Bitcoin o Ethereum.



CATEGORÍAS SISTEMAS DISTRIBUIDOS

1. **Cluster computing:** Cuando todas las máquinas están en la misma ubicación en una red local (típicamente en una sala de servidores), quiere decir, las máquinas están ubicadas en la misma dirección como la empresa ocupándolas.
2. **Cloud computing:** Cuando todas las máquinas están en una ubicación central, pero no local. Esto quiere decir que existe un clúster (como en el caso anterior), pero el clúster no está en la misma ubicación física como la empresa, y los usuarios se conectan al clúster de manera remota para ejecutar sus tareas. El ejemplo de esto será el Amazon Web Services, qué maneja varios clústeres a los cuales uno puede conectarse.
3. **Grid computing:** Cuando las máquinas están en distintas ubicaciones, pero no están todas juntas como en el caso de la nube. Este tipo de cómputo se usa principalmente en proyectos científicos como SETI, o en criptomonedas como Bitcoin.

DISTRIBUCIÓN DE LOS DATOS

En termino de manejo de los datos, un sistema distribuido para manejo de datos no solo permite guardar más datos usando más máquinas/servidores, sino también permite mejorar varios aspectos en el flujo de información de una empresa. Por ejemplo, un sistema distribuido va a mantener varias rutas para acceder a los datos a través de distintas máquinas. En nuestro ejemplo de WebShop, esto significaría que la empresa va a tener más de un servidor con la página web de la empresa. La gracia de esto es qué si un servidor con la página web se desconecta, o tiene una falla técnica, el usuario se puede conectar a su cuenta de WebShop a través de otro servidor. Esto quiere decir que el acceso a los datos mejoró. Similarmente, una organización como WebShop puede tener sedes en varias ciudades del país. Por este motivo, la gente de la unidad de despachos en una ciudad probablemente va a necesitar solo los datos de los clientes viviendo en esta ciudad, y no la información de todos los clientes de WebShop. Por esto, en sistemas distribuidos uno puede distribuir los datos basado en la localidad en la cual se utilizan.

REPLICACIÓN Y FRAGMENTACIÓN DE LOS DATOS

Pensando en una empresa grande que mantiene una tabla con información de sus usuarios como, por ejemplo, Amazon o Google, ya sabemos que todos estos datos no van a caber en un solo computador, y tienen que ser distribuidos. La pregunta es entonces ¿qué técnica uno puede ocupar para distribuir estos datos en varias máquinas? Para simplificar las cosas, consideremos el caso cuando los datos de los usuarios se guardan en una tabla llamada Usuarios, donde el userID es la llave de la relación. La tabla tendría el siguiente formato:

Usuarios

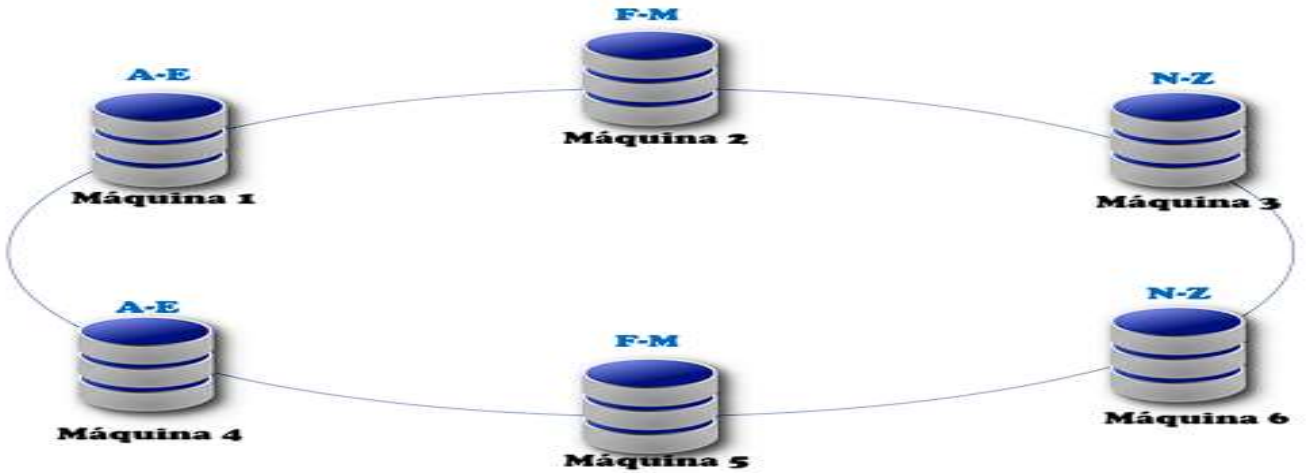
userID	Nombre	email	Ciudad	...
PM123	Pedro Morales	pmorales@uc.cl	Santiago	
gArenas	Gonzalo Arenas	arenas1@gmail.com	Santiago	
...	...	...	...	

FRAGMENTACIÓN

Consiste en dividir la relación en varios pedazos, guardando cada pedazo en distinto servidor. Por ejemplo, si tenemos una relación como Usuarios de arriba, de tamaño muy grande, podemos dividir la relación agrupando todos los usuarios cuyo userID empieza con una letra entre A-E, todos los usuarios cuyo userID empieza con letra F-M, y todos los usuarios cuyo userID empieza con la letra N-Z (aquí estamos asumiendo que el userID siempre empieza con una letra). De esta forma podemos guardar cada grupo en un servidor distinto. Se usa mucho para distribuir los datos basado en localidad, por ejemplo, guardando datos sobre los clientes en una ciudad en el servidor que está ubicado en esta misma ciudad.

REPLICACIÓN

Guarda los mismos datos en distintos servidores. Tener datos replicados en varios lugares tiene muchas ventajas. Primero, si la máquina en un lugar falla o se desconecta, uno todavía puede acceder a mismos datos. Esto permite recuperar a los datos, pero también hace que el usuario final no necesite esperar mucho tiempo para acceder a los datos en el caso de fallas. También, como muchos usuarios finales ejecutan consultas similares sobre mismos datos (como, por ejemplo, los clientes de un banco revisando sus cuentas), tener la copia de los datos en distintos servidores permite ejecutar las consultas más rápido, y en paralelo.



VENTAJAS SISTEMAS DISTRIBUIDOS

- Mejoras en el volumen de datos que pueden guardar y la eficiencia en procesar estos datos.
- Son fáciles para extender, porque normalmente es suficiente conectar más máquinas a la red distribuida.
- Protección de los datos valorables es más fácil, porque los datos son replicados en varios sitios.
- Siguen funcionando, incluso cuando algunas máquinas fallan o se desconectan de la red.

DESVENTAJAS SISTEMAS DISTRIBUIDOS

- Son sistemas mucho más complejos, y requieren gente especializada para desarrollarlos y mantenerlos.
- Son generalmente más caros, tanto en términos de máquinas, como en término de labor necesario para mantenerlos.
- Como las bases de datos distribuidas son una tecnología relativamente nueva, todavía no hay mucha estandarización de cómo implementarlas, ni muchos profesionales capaces de mantenerlas.
- Dado muchos sitios donde uno guarda los datos, pueden tener más vulnerabilidades.

BIBLIOGRAFÍA

- Ramakrishnan, R., Gehrke, J., Database Management Systems, 3rd edition, McGraw-Hill, 2002.