Tarea 1. Bases de datos distribuidas

1. Concepto general de una base de datos distribuida

Una base de datos distribuida es un sistema de gestión de bases de datos en el que las bases de datos no están ubicadas en un sistema de servidor común. Las bases de datos pueden almacenarse en varios servidores de datos ubicados en la misma ubicación física, o pueden distribuirse a través de una red de área local o amplia.

2. Diferencia entre BD en la nube y BD distribuida

Base de datos en la nube

Servicios de bases de datos diseñados y accesibles a traves de plataformas de computacion en la nube, estas bases ofrecen funcionalidades similares a las tradicionales, pero con la flexibilidad añadida del entorno en la nube. Los usuarios pueden instalar software en una infraestructura en la nube para implementar la base de datos permitiendo el acceso desde cualquier lugar. Además, facilitan la escalabilidad, ya que pueden ampliar la capacidad de almacenamiento en tiempo real.

Base de datos distribuida

Sistemas de bases de datos en los que los datos se almacenan en múltiples ubicaciones físicas, ya sea en diferentes servidores o dentro de un mismo centro de datos o en distintos centros de datos geográficamente dispersos. La principal característica de esta base de datos es que permite que las operaciones se realicen de manera local en cada nodo, mejorando la eficiencia y reduciendo la latencia.

3. Arquitectura de Computadoras

Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es uno de los modelos más tradicionales en bases de datos distribuidas. Consiste en uno o varios servidores que almacenan y gestionan los datos, mientras que los clientes acceden a esos datos a través de una red. Los clientes realizan solicitudes que el servidor procesa y responde, manteniendo una estructura centralizada del sistema (Özsu & Valduriez, 2020).

Este modelo facilita el control de acceso, la seguridad y la administración de los datos, ya que todo se gestiona desde el servidor. Además, permite la actualización de los servicios sin afectar a los clientes. Sin embargo, presenta limitaciones como la dependencia del servidor central, que puede convertirse en un punto único de falla si se sobrecarga o sufre una interrupción (fs.com, 2022).

La arquitectura cliente-servidor es común en entornos corporativos y en aplicaciones como correos electrónicos o servicios en la nube, donde se requiere consistencia y control centralizado (Arsys, 2023). Su diseño permite escalabilidad y mantenimiento eficiente, aunque a un mayor costo técnico y operativo.

Arquitectura Peer to Peer (P2P)

En la arquitectura peer-to-peer (P2P), todos los nodos tienen funciones equivalentes, actuando como clientes y servidores simultáneamente. No existe un nodo central que gestione los recursos; en su lugar, cada nodo puede compartir datos directamente con otros. Esto proporciona una gran escalabilidad y tolerancia a fallos (Coulouris et al., 2011).

Al eliminar el servidor central, se reducen los cuellos de botella y se mejora la disponibilidad. Sin embargo, también implica desafíos en la seguridad, sincronización y respaldo de datos, ya que la responsabilidad de estas tareas recae en cada nodo individual (fs.com, 2022).

Esta arquitectura se emplea en aplicaciones como redes de compartición de archivos, plataformas descentralizadas y algunas redes blockchain. Existen variantes híbridas, como el caso de Napster, que utilizaba un servidor para indexar archivos pero distribuía el contenido entre los usuarios (reactiveprogramming.io, 2022).

Arquitectura Jerárquica o Multinivel

La arquitectura multinivel organiza los nodos de forma jerárquica, permitiendo estructurar la red en diferentes niveles de control. Los nodos superiores coordinan a los subordinados, facilitando la gestión local de datos y su posterior sincronización con un sistema global (Elmasri & Navathe, 2015).

Este enfoque es útil en organizaciones distribuidas geográficamente, como bancos o empresas multinacionales, donde se requiere autonomía local y coordinación central. Reduce la cantidad de tráfico en la red y permite manejar diferentes niveles de acceso a la información (fs.com, 2022).

Sin embargo, esta estructura puede volverse compleja de configurar y mantener, especialmente al definir políticas de replicación y comunicación entre niveles. Además, los nodos en los niveles más altos representan puntos críticos cuya falla puede afectar a varias unidades subordinadas.

4. Estrategias para distribuir una base de datos

Fragmentación de Datos

La fragmentación consiste en dividir una base de datos en partes más pequeñas llamadas fragmentos, que pueden ser distribuidos en diferentes nodos de una red. Existen principalmente dos tipos:

- Fragmentación horizontal: divide las tablas en filas, asignando subconjuntos de registros a distintos nodos.
- Fragmentación vertical: divide las tablas en columnas, distribuyendo diferentes atributos entre nodos.

Esta estrategia mejora el rendimiento al permitir que las consultas se ejecuten localmente en los nodos que contienen los datos relevantes, reduciendo la necesidad de transferencias de datos a través de la red. Sin embargo, requiere un diseño cuidadoso para mantener la integridad y coherencia de los datos. (Wikipedia)

Replicación de Datos

La replicación implica almacenar copias de los datos en múltiples nodos. Puede ser:

- Replicación completa: todos los datos se replican en cada nodo.
- Replicación parcial: solo una parte de los datos se replica en ciertos nodos.

Esta estrategia aumenta la disponibilidad y tolerancia a fallos, ya que, si un nodo falla, otros pueden proporcionar los datos. Sin embargo, introduce desafíos en la sincronización y mantenimiento de la coherencia entre las copias, especialmente en sistemas con actualizaciones frecuentes. (ArcGis Pro)

Particionamiento de Datos (Sharding)

El particionamiento, o sharding, divide los datos en fragmentos que se almacenan en diferentes nodos. Las estrategias comunes incluyen:

- Particionamiento por rango: los datos se dividen según un rango de valores de una clave.
- Particionamiento por hash: se aplica una función hash a una clave para determinar la ubicación del fragmento. (Dev Comunity, 2022)
- Particionamiento geográfico: los datos se distribuyen según la ubicación geográfica de los usuarios o nodos.

El particionamiento mejora la escalabilidad y el rendimiento, permitiendo que las consultas se dirijan directamente al nodo que contiene los datos relevantes. Sin embargo, puede complicar las consultas que requieren datos de múltiples fragmentos. (LinkedIn, 2023)

Estrategias de Diseño: Enfoques Top-Down y Bottom-Up

Al diseñar una base de datos distribuida, se pueden adoptar dos enfoques:

- Enfoque Top-Down: se inicia con un diseño global del sistema y luego se distribuyen los datos según las necesidades identificadas.
- Enfoque Bottom-Up: se parte de bases de datos existentes en diferentes nodos y se integran en un sistema distribuido.

La elección del enfoque depende de factores como la existencia de sistemas previos, los requisitos de la organización y la complejidad del sistema. (Pure Storage)

5. Gestores y funciones para Bases de Datos Distribuidas (replicación)

Características y Funciones (SQL)

Las bases de datos relacionales distribuidas permiten que múltiples sistemas gestionen y accedan a datos de manera coordinada. Cada sistema posee su propio gestor de base de datos, como Db2 de IBM, que se comunica con otros para ofrecer una vista unificada de los datos. Esto facilita que las aplicaciones utilicen SQL para acceder a datos distribuidos en diferentes sistemas interconectados. I

Mecanismos de Replicación

La replicación en bases de datos relacionales puede ser:

- Síncrona: Las transacciones se confirman en todos los nodos simultáneamente, asegurando consistencia inmediata.
- Asíncrona: Las actualizaciones se propagan a los nodos secundarios después de confirmarse en el nodo principal, lo que puede introducir cierta latencia pero mejora el rendimiento.
- Semisíncrona: Combina aspectos de ambos métodos, buscando un equilibrio entre consistencia y rendimiento.

Ejemplos de Gestores Relacionales Distribuidos

- IBM Db2: Soporta bases de datos relacionales distribuidas, permitiendo acceso a datos en sistemas interconectados mediante SQL. IBM - United States
- Oracle Database: Ofrece características avanzadas para entornos distribuidos, incluyendo replicación y alta disponibilidad.
- Microsoft SQL Server: Proporciona soluciones de replicación y sincronización para bases de datos distribuidas.

Características y Funciones (NoSQL)

Las bases de datos NoSQL están diseñadas para manejar grandes volúmenes de datos no estructurados o semiestructurados, ofreciendo alta escalabilidad y flexibilidad. Utilizan diversos modelos de datos, como documentos, clave-valor, columnas y grafos. Estas bases de datos son ideales para aplicaciones que requieren baja latencia y alta disponibilidad.

Mecanismos de Replicación

La replicación en bases de datos NoSQL implica copiar y almacenar datos en múltiples servidores, lo que proporciona:

- Alta disponibilidad: Permite el acceso continuo a los datos incluso durante fallos de servidores.
- Redundancia: Protege contra la pérdida de datos al mantener múltiples copias.
- Escalabilidad: Facilita el manejo de grandes volúmenes de datos distribuidos geográficamente.
- Ejemplos de Gestores NoSQL Distribuidos
- MongoDB: Base de datos orientada a documentos que permite la replicación y partición de datos para escalabilidad horizontal.
- Couchbase Server: Ofrece replicación de datos, reconfiguración en vivo del clúster y soporte para múltiples modelos de datos.
- Apache Cassandra: Diseñada para manejar grandes cantidades de datos distribuidos en múltiples servidores, con replicación automática y tolerancia a fallos.

Referencias

- Özsu, M. T., & Valduriez, P. (2020). *Principles of Distributed Database Systems (4th ed.)*. Springer. http://5.202.73.55:8026/opac/temp/7593.pdfhttp://5.202.73.55:8026/opac/temp/7593.pdf
- Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed Systems: Concepts and Design* (5th ed.). Addison-Wesley. https://ftp.utcluj.ro/pub/users/civan/CPD/3.RESURSE/6.Book_2012_Distributed%20systems%20_Couloris.pdfhttps://ftp.utcluj.ro/pub/users/civan/CPD/3.RESURSE/6.Book_2012_Distributed%20systems%20_Couloris.pdf
- Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems (7th ed.)*. Pearson. http://ir.harambeeuniversity.edu.et/bitstream/handle/123456789/1810/Fundamentals%20of%20Database%20Systems%20.pdf%20%28%20PDFDrive.com%20%29.pdf?sequence=1&isAllowed=yhttp://ir.harambeeuniversity.edu.et/bitstream/handle/123456789/1810/Fundamentals%20of%20Database%20Systems%20.pdf%20%28%20PDFDrive.com%20%29.pdf?sequence=1&isAllowed=y
- Arsys. (2023). *Todo sobre la arquitectura cliente-servidor*. https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor
- fs.com. (2022). *Client-Server vs Peer-to-Peer Networks*. https://www.fs.com/es/blog/client-server-vs-peer-to-peer-networks-5464.html
- reactiveprogramming.io. (2022). *Arquitectura P2P*. https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/p2p
- ArcGIS Pro. (s.f.). *Introducción a los datos distribuidos*. Esri. https://pro.arcgis.com/es/pro-app/latest/help/data/geodatabases/overview/introduction-to-distributed-data.htm
- Dev Community. (2022). *Bases de datos distribuidas: sharding*. https://dev.to/anfibiacreativa/bases-de-datos-distribuidas-sharding-4a39
- LinkedIn. (2023). *Estrategias de partición de datos para sistemas distribuidos*. https://www.linkedin.com/advice/1/what-some-common-data-partitioning-strategies-tribe?lang=es
- Wikipedia. (s.f.). *Base de datos distribuida*.
- Pure Storage. (s/f). *¿Qué es el procesamiento de datos distribuidos?* https://www.purestorage.com/es/knowledge/what-is-distributed-data-processing.html
- *IBM Db2 12 - Introducción - Bases de datos relacionales distribuidas: https://www.ibm.com/docs/es/db2-for-zos/12?topic=concepts-distributed-relational-databases*
- ***Replicación de datos: Tipos y beneficios explicados - Astera**: https://www.astera.com/es/type/blog/data-replication/ IBM - United States+5Astera+5IBM - United States+5*
- ***Replicación de bases de datos: Técnicas y ventajas - StudySmarter ES**: https://www.studysmarter.es/resumenes/ciencias-de-la-computacion/bases-de-datos/replicacion-de-bases-de-datos/ StudySmarter ES*
- *IBM. (s.f.). ¿Qué es una base de datos en la nube?. IBM. Recuperado el 2 de mayo de 2025, de https://www.ibm.com/mx-es/topics/cloud-database*
- *Mosquera, J. (2023). Guía 2023: Bases de Datos Distribuidas - Uso y Ventajas. JhonMosquera.com. Recuperado el 2 de mayo de 2025, de https://jhonmosquera.com/bases-de-datos-distribuidas/*
- *IBM. (s.f.). Diseño de bases de datos distribuidas. IBM Documentation. Recuperado el 2 de mayo de 2025, de https://www.ibm.com/docs/es/db2/11.1?topic=managers-designing-distributed-databases*