



UNIVERSIDAD AUTÓNOMA DE CHIAPAS.

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN, CAMPUS I.

LICENCIATURA EN INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS DE
SOFTWARE.

OCTAVO SEMESTRE, GRUPO: "M"

MATERIA: COMPUTO DISTRIBUIDO

DOCENTE: MTRO. NAÑEZ COUTIÑO ADAN.

ALUMNO: CARLOS DANIEL AMORES HERNANDEZ

"ACTIVIDAD PRELIMINAR: BASES DE DATOS DISTRIBUIDAS TAREA"

FECHA DE ENTREGA: 5 DE MARZO DE 2025.

¿Que es una base de datos distribuida?

Es una base de datos que no está restringida a un solo sistema y está dispersa en numerosos lugares, como dos o más computadoras o una red de computadoras. Una base de datos distribuida o un sistema de gestión de datos se distribuye en varios sitios sin partes físicas. Esto puede ser necesario si una base de datos específica debe ser accesible para muchas personas en todo el mundo. Por lo tanto, debe administrarse de manera que se presente a los consumidores como una única base de datos.

Las bases de datos distribuidas se pueden utilizar para la escalabilidad horizontal y para satisfacer las necesidades de carga sin cambiar el esquema de la base de datos ni hacer crecer verticalmente un solo sistema. Asimismo las bases de datos distribuidas abordan preocupaciones que pueden surgir al utilizar un solo sistema y una sola base de datos, incluida la disponibilidad, la tolerancia a fallas, el rendimiento, la latencia, la escalabilidad y muchas más.

Las características principales de las bases de datos distribuidas (BDD) :

1. Autonomía Local: Cada sitio puede controlar sus propios datos, administrar la seguridad y manejar transacciones de manera independiente.
2. Disponibilidad y Tolerancia a Fallos: Si un sitio pierde conectividad, los demás pueden seguir funcionando sin afectar toda la red.
3. Seguridad y Cumplimiento: Permiten aplicar restricciones específicas según la ubicación, como cumplir con regulaciones como el GDPR.
4. Métodos de Almacenamiento:
 - a. Duplicación: Replica datos en varias ubicaciones para mejorar disponibilidad y consultas en paralelo, aunque requiere sincronización constante.
 - b. Fragmentación: Divide los datos en partes más pequeñas para evitar duplicaciones y mantener la coherencia, con modalidades:
 - i. Horizontal: Separa filas en diferentes fragmentos.
 - ii. Vertical: Divide los esquemas en subconjuntos más pequeños.
5. Tipos de Bases de Datos Distribuidas:
 - a. Homogéneas: Usan el mismo DBMS y sistemas operativos en todos los sitios, con comunicación fluida entre ellos.
 - b. Heterogéneas: Integran diferentes DBMS, sistemas operativos y modelos de datos, lo que puede complicar la ejecución de consultas y transacciones.
6. Escalabilidad y Rendimiento: Facilitan la gestión de grandes volúmenes de datos y permiten aumentar la capacidad del sistema sin afectar el rendimiento.
7. Reducción de Costos: Disminuyen la necesidad de servidores centralizados y costoso mantenimiento.
8. Casos de Uso: Son utilizadas por grandes empresas tecnológicas como Google, Amazon, Microsoft, bancos y servicios en la nube debido a la gran cantidad de datos que manejan.

Tipos de Datos Distribuidos

Tipos de datos distribuidos

Los datos distribuidos se pueden dividir en cinco tipos básicos, como se detalla a continuación:

Datos duplicados: la replicación de datos se usa para crear instancias adicionales de datos en diferentes partes de la base de datos. Usando esta táctica, una base de datos distribuida puede evitar el tráfico excesivo porque se puede acceder a los datos idénticos localmente.

Datos fragmentados horizontalmente: esta categoría de distribución de datos implica el uso de claves primarias (cada una de las cuales se refiere a un registro en la base de datos). La fragmentación horizontal se usa comúnmente para situaciones en las que ubicaciones específicas de una empresa generalmente solo necesitan acceso a la base de datos correspondiente a su rama específica.

Datos verticalmente fragmentados: con la fragmentación vertical, las claves primarias se utilizan nuevamente. Sin embargo, en este caso, las copias de la clave primaria están disponibles dentro de cada sección de la base de datos (accesible para cada rama). Este tipo de formato funciona bien para situaciones en las que una sucursal de una empresa y la ubicación central interactúan con las mismas cuentas, pero quizás de diferentes maneras (como cambios en la información de contacto del cliente frente a cambios en las cifras financieras).

Datos reorganizados: la reorganización significa que los datos se han ajustado de una manera u otra, como es típico en las bases de datos de soporte de decisión. En algunos casos, hay dos sistemas distintos que manejan las transacciones y el soporte de decisiones.

Datos de esquema separado: esta categoría de datos divide la base de datos y el software utilizado para acceder a diferentes departamentos y situaciones, por ejemplo, datos de usuario frente a datos de productos. Por lo general, existe una superposición entre las diversas bases de datos dentro de este tipo de distribución.

Importancia de una BDD

Las bases de datos distribuidas son fundamentales porque:

- Mejoran la disponibilidad: Si un nodo falla, otros pueden seguir operando.
- Incrementan el rendimiento: Permiten realizar consultas y transacciones en paralelo.
- Reducen la latencia: Almacenan datos en diferentes ubicaciones cercanas al usuario.
- Garantizan escalabilidad: Pueden crecer añadiendo más nodos sin afectar el sistema.
- Aseguran tolerancia a fallos: Distribuyen datos en múltiples servidores para evitar pérdidas de información.

- Cumplen con regulaciones: Permiten aplicar normas como el GDPR en diferentes regiones.

Teorema CAP y Subdivisiones de los SAD (Sistemas de Almacenamiento Distribuido)

El Teorema CAP establece que en un sistema distribuido solo se pueden garantizar dos de las siguientes tres propiedades al mismo tiempo:

1. Consistencia (C): Todos los nodos tienen los mismos datos en todo momento.
2. Disponibilidad (A): El sistema responde siempre, incluso si algunos nodos fallan.
3. Tolerancia a Particiones (P): El sistema sigue funcionando aunque haya problemas de comunicación entre nodos.

Ejemplo:

- CP (Consistencia + Tolerancia a Particiones) → BigTable (Google), HBase (Hadoop)
- AP (Disponibilidad + Tolerancia a Particiones) → Cassandra, DynamoDB (Amazon)
- CA (Consistencia + Disponibilidad) → No es posible en sistemas distribuidos reales, solo en bases de datos centralizadas.

Subdivisiones de los SAD

1. Sistemas de Archivos Distribuidos: Manejan grandes volúmenes de datos en múltiples servidores.
 - a. Ejemplo: Google File System (GFS), Hadoop Distributed File System (HDFS).
2. Bases de Datos Distribuidas: Gestionan datos estructurados en varias ubicaciones.
 - a. Ejemplo: MongoDB, Cassandra, MySQL Cluster.
3. Almacenamiento en la Nube: Infraestructura distribuida en la nube pública o privada.
 - a. Ejemplo: Google Cloud Storage, Amazon S3, Azure Blob Storage.

Características de ACID y BASE

ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)

Propiedades que garantizan confiabilidad en las bases de datos tradicionales:

1. Atomicidad: Una transacción se completa en su totalidad o no se ejecuta en absoluto.
 - a. Ejemplo: Una transferencia bancaria debe descontar dinero de una cuenta y sumarlo a otra, sin quedar en estado intermedio.
2. Consistencia: Los datos siempre permanecen en un estado válido después de una transacción.
3. Aislamiento: Las transacciones concurrentes no afectan su ejecución.
4. Durabilidad: Una vez confirmada una transacción, los cambios quedan guardados permanentemente, incluso si hay fallos en el sistema.

Ejemplo de uso: Bases de datos relacionales como MySQL, PostgreSQL, Oracle.

BASE (Basically Available, Soft-state, Eventually Consistent)

Propiedades usadas en bases de datos NoSQL y sistemas distribuidos escalables:

- Básicamente Disponible: Garantiza disponibilidad aunque algunos nodos puedan fallar.
- Soft-state: El estado del sistema puede cambiar con el tiempo sin requerir una actualización inmediata.
- Eventual Consistencia: Los datos pueden no estar sincronizados en todos los nodos, pero eventualmente llegarán a un estado consistente.

Ejemplo de uso: Bases de datos NoSQL como Cassandra, DynamoDB, CouchDB.