

Una verdadera base de datos distribuida (DDB) es aquella que presenta un esquema descentralizado de administración de datos donde los archivos están esparcidos a través de una colección de nodos autónomos y que se comunican a través de un lenguaje común. De manera que almacene, actualice y recupere los datos independientemente del usuario. Los datos y configuración son transparentes a los usuarios, de manera que estén disponibles en un nodo cercano al usuario. Además, los nodos pueden ser añadidos y borrados sin afectar el acceso y uso de datos. El artículo introduce soluciones a los problemas de las DDB:

**Diseño de la distribución:** Se presentan dos alternativas de diseño: Descentralizar por función cuando la información es accedida por los mismos usuarios y descentralizar por ubicación cuando se busca particionar información aprovechando la localidad de los datos. Otro punto es el grado de descentralización. Por ejemplo, mantenimiento de nodos descentralizado y control de red centralizado.

**Búsquedas:** Uno de problemas es el desconocimiento centralizado de toda la DDB. Es útil clasificar a los datos de la siguiente manera:

- Datos globales: información compartida en todos los sitios (réplicas)
- Datos locales: información relevante a un solo sitio, pero accesible para todos (particionamiento)
- Semiglobal es internodal y a veces intersitio.

**Actualizaciones:** El particionamiento de datos es más eficiente ante actualizaciones frecuentes, con operaciones de lectura no locales son más costosas. En cambio, los datos replicados son más eficientes cuando hay múltiples lecturas de datos, pero actualizaciones menos frecuentes.

**Relacional:** Expandir una base de datos relacional implica añadir nuevas tablas o entradas que se ligan lógicamente por el software de gestión de datos: Los datos no son dependientes de otros datos siendo adecuados para un ambiente distribuido

**Integridad de datos:** Las propiedades para conservar la integridad de los datos son:

- Consistencia: Cada transacción debe terminar en un estado válido para la DDB
- Atómico: La transacción es terminada por completo o anulada
- Durable: Una vez que la transacción terminó, no puede ser anulada sin una transacción inversa

Para mantener integridad se debe asegurar que los mensajes entre nodos lleguen correctamente: "two phase commit"

**Transacciones actualizadas:** El "two phase commit protocol" usa un programa coordinador para centralizar las decisiones de actualización o abortar una transacción. Tiene comunicación con todos los participantes de la transacción donde cada uno entra en una fase de preparación en la que actualiza o aborta su parte. Después transmiten un mensaje al coordinador el cual indica a todos hacen actualización o abortan. Al final se envía una confirmación a todos.

**Administración distribuida:** Algunos aspectos comunes a toda la red deben ser designados y controlados centralmente. Por otra parte, las funciones de la base de datos locales deben ser controlados por el nodo local.

**Selección de DDB:** Se debe considerar la flexibilidad en la red para acceder a cambios cuando se requieran. Además, proveer transparencia entre datos y usuarios al mismo tiempo que se asegura la integridad. Por último, mantener disponibilidad ante fallas de componentes o con inaccesibilidad temporal.