Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

los componentes de una base de datos distribuidas eh

Son el **sistema gestor local de la base de datos** que funciona como un sistema estándar de administración de la base de datos responsable de controlar los datos locales en cada sitio que tenga la base base de datos cuenta con su propio diccionario de datos locales así como los subsistemas usuales para el control de concurrencia recuperación y optimización de consulta

**el sistema gestor distribuido de la base** de datos implica un conjunto de programas que operan en diversas computadoras y puede estar distribuido en los servidores locales estos programas pueden ser subsistemas de un único gestor de un mismo fabricante O podría consistir en una colección de programas de diferentes proveedores

**una red de comunicaciones** es el software en cada nodo que lo liga con una red eh que incluye la descripción completa de los nodos y las líneas de red de igual manera identifica para cada nodo el procesamiento realizado la capacidad de almacenamiento Potencia de procesamiento y su estado actual identifica para cada vínculo los nodos que conecta el tipo de vínculo el ancho de B los protocolos requeridos y su estado actual

el **diccionario de datos** global es una recopilación de información acerca de la base de datos distribuida incluye una lista de todos los aspectos de sus datos bajo la forma de metadatos su ubicación y otra clase de información sobre cualquier dato almacenado en cualquier parte del sistema distribuido eh los diccionarios pueden ser de tipo centralizado es Global en una única sede el inconveniente es que se transforma en un cuello de botella o descentralizado es local en cada nodo el inconveniente Es que la búsqueda no local es costosa y el esquema mixto es Global en un único nodo y local en cada sede.

un sistema de bases de datos distribuida eh consiste en un conjunto de bases de datos lógicamente relacionadas y distribuidas sobre una red de computadoras interconectadas que se encuentran en varios lugares geográficos distintos y se administran de forma local y global por ejemplo un sistema bancario compuesto por cuatro sucursales situadas en cuatro ciudades diferentes cada sucursal posee su propia computadora con una base de datos que alberga todas las cuentas abiertas en dicha sucursal así cada una de estas instalaciones se considera un sitio también hay un sitio único que mantiene la información relativa a todas las sucursales del banco eh Hay varios factores que han hecho que las bases de datos evolucionen a bases de datos distribuidas en el mundo de los negocios se ha dado el fenómeno de la globalización y a la vez los negocios de las empresas son cada vez más descentralizados geográficamente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

los datos en las bases de datos distribuidas deben contar con la característica de Independencia esa Independencia puede ser tanto lógica como física

**la independencia lógica** de datos se refiere a la inmunidad de las aplicaciones de usuario a los cambios en la estructura lógica de la base de datos permitiendo que un cambio en la definición de un esquema no afecte a las aplicaciones del usuario por ejemplo Agregar un nuevo atributo a la relación eh la creación de una nueva relación el reordenamiento lógico de algunos atributos etcétera

**eh la independencia física** de los datos se refiere al ocultamiento de los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario la descripción física de datos puede cambiar sin afectar a las aplicaciones por ejemplo los datos pueden ser movidos de un disco a otro o la organización de los datos en el disco puede cambiar.

Texto

Descripción generada automáticamente

otra característica de las bases distribuidas es la transparencia su objetivo es ocultar al usuario los detalles de diseño es decir el usuario no tiene que saber que se encuentra trabajando con un sistema distribuido mediante independencia de datos debe asegurar que dos sitios no utilicen el mismo nombre para elementos de datos diferentes una sol a este problema es exigir que todos los nombres se registren en un servidor de nombres central un enfoque alternativo muy utilizado propone que cada sitio anteponga su propio identificador de sitio a cualquier nombre que genere garantizando así que dos sitios diferentes no generen nunca el mismo nombre dado que cada sitio tiene un identificador único.

Diagrama

Descripción generada automáticamente con confianza media

el propósito de establecer lectura de un sistema de bases de datos distribuida es ofrecer un nivel de transparencia adecuado para el manejo de la información para ello se estructura por niveles a partir de la independencia de datos

**el primer nivel** soporta la transparencia de red que es la ocultación al usuario de los componentes de este sistema distribuido y están separados por lo tanto el usuario percibirá que el sistema es un único sistema y no varios componentes el

**segundo nivel permite** la transparencia de replicación de datos y se refiere a que si existen réplicas de datos estas deben ser controladas por el sistema gestor no por el usuario la función principal de la transparencia de replicación es la de mantener la consistencia entre las copias la replicación es necesaria para obtener un mayor rendimiento debido a que se dispone de copias locales y mayor disponibilidad ya que los datos son accesibles siempre al tener varias copias eh la **principal desventaja** es que hay que mantener actualizadas todas las copias de ese dato replicado y esto lleva al problema de la denominada **propagación de las actualizaciones**

**tercer nivel** permite la transparencia de fragmentación el acceso a la base de datos es de forma global es decir el usuario no necesita especificar los nombres de los fragmentos ni las ubicaciones de los datos los usuarios deben comportarse como si los datos en realidad no estuvieran distribuidos el sistema maneja la conversión de consultas de usuarios definidas sobre relaciones globales a consultas definidas sobre fragmentos las respuestas a consultas fragmentadas resultan en una sola respuesta global

**el cuarto nivel permite la** transparencia de acceso o localización por medio de un lenguaje de manipulación de datos le permite Acceder al usuario a los datos sin tener en cuenta la ubicación de estos no necesita saber dónde está ese dato para utilizarlo eh se consigue cuando los administradores de transacciones distribuidas pueden determinar la localización de los datos y emitir acciones a los administradores apropiados esto se puede ejecutar cuando los administradores de transacciones tienen acceso a los directorios de localización de datos los administradores de transacciones los sistemas necesitan conocer si los datos cambian de lugar ya que las transacciones ignoran la modificación en la localización.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteTexto

Descripción generada automáticamente

el propósito de establecer lectura de un sistema de bases de datos distribuida es ofrecer un nivel de transparencia adecuado para el manejo de la información para ello se estructura por niveles a partir de la independencia de datos

**el primer nivel** soporta la transparencia de red que es la ocultación al usuario de los componentes de este sistema distribuido y están separados por lo tanto el usuario percibirá que el sistema es un único sistema y no varios componentes el

**segundo nivel permite** la transparencia de replicación de datos y se refiere a que si existen réplicas de datos estas deben ser controladas por el sistema gestor no por el usuario la función principal de la transparencia de replicación es la de mantener la consistencia entre las copias la replicación es necesaria para obtener un mayor rendimiento debido a que se dispone de copias locales y mayor disponibilidad ya que los datos son accesibles siempre al tener varias copias eh la **principal desventaja** es que hay que mantener actualizadas todas las copias de ese dato replicado y esto lleva al problema de la denominada **propagación de las actualizaciones**

**tercer nivel** permite la transparencia de fragmentación el acceso a la base de datos es de forma global es decir el usuario no necesita especificar los nombres de los fragmentos ni las ubicaciones de los datos los usuarios deben comportarse como si los datos en realidad no estuvieran distribuidos el sistema maneja la conversión de consultas de usuarios definidas sobre relaciones globales a consultas definidas sobre fragmentos las respuestas a consultas fragmentadas resultan en una sola respuesta global

**el cuarto nivel permite la** transparencia de acceso o localización por medio de un lenguaje de manipulación de datos le permite Acceder al usuario a los datos sin tener en cuenta la ubicación de estos no necesita saber dónde está ese dato para utilizarlo eh se consigue cuando los administradores de transacciones distribuidas pueden determinar la localización de los datos y emitir acciones a los administradores apropiados esto se puede ejecutar cuando los administradores de transacciones tienen acceso a los directorios de localización de datos los administradores de transacciones los sistemas necesitan conocer si los datos cambian de lugar ya que las transacciones ignoran la modificación en la localización.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Un problema fundamental en las bases de datos distribuidas es cómo guardar los datos de manera redundante, de manera que incluso si falla un nodo de la arquitectura en particular, los datos se puedan recuperar a partir de otras fuentes. A partir de una relación *r* o tabla, es posible utilizar las siguientes técnicas:

* **Replicación**: El sistema mantiene varias réplicas idénticas de una relación *r* o tabla. Cada réplica se almacena en un sitio diferente.
* **Fragmentación**: Una relación se particiona en varios fragmentos y cada fragmento es almacenado en un sitio diferente.
* **Combinación de replicación y fragmentación**: Una relación es particionada en varios fragmentos y el sistema mantiene varias copias de estos fragmentos.

Diagrama

Descripción generada automáticamente

Un problema fundamental en las bases de datos distribuidas es cómo guardar los datos de manera redundante, de manera que incluso si falla un nodo de la arquitectura en particular, los datos se puedan recuperar a partir de otras fuentes. A partir de una relación *r* o tabla, es posible utilizar las siguientes técnicas:

* **Replicación**: El sistema mantiene varias réplicas idénticas de una relación *r* o tabla. Cada réplica se almacena en un sitio diferente.
* **Fragmentación**: Una relación se particiona en varios fragmentos y cada fragmento es almacenado en un sitio diferente.
* **Combinación de replicación y fragmentación**: Una relación es particionada en varios fragmentos y el sistema mantiene varias copias de estos fragmentos.

Diagrama

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

Mediante la técnica de réplicas, el sistema conserva copias idénticas de la relación y guarda cada réplica en un sitio diferente.

La replicación de datos se usa para crear instancias adicionales de datos en diferentes partes de la base.

Usando esta táctica, una base de datos distribuida puede evitar el tráfico excesivo porque se puede acceder a datos idénticos en forma local y facilita el suministro de datos desde cualquier sección a otra si los datos de esta última sección se ven comprometidos por cualquier tipo de error.

En general, la réplica mejora el rendimiento de las operaciones de lectura y aumenta la disponibilidad de los datos.

Sin embargo, las transacciones de actualización suponen una mayor sobrecarga: el sistema debe asegurar que todas las réplicas de la relación sean consistentes.

Cada actualización sobre la relación o tabla debe ser propagada a los sitios que contienen las respectivas réplicas. El control de las actualizaciones concurrentes de los datos replicados resulta más complejo que en los sistemas centralizados.

En las bases de datos distribuidas, las actualizaciones pueden configurarse según la necesidad de los datos correctos en un momento dado o durante cualquier periodo de tiempo, pudiendo asumir los siguientes formatos: en forma periódica o en forma continua.

En la réplica periódica, los datos en la réplica se actualizan y se sincronizan con el repositorio principal solo en períodos determinados de tiempo. Este enfoque se utiliza cuando no hay una necesidad urgente de disponer de los datos actualizados de forma inmediata.

Cuando es muy necesario disponer de los datos actualizados en los diferentes sitios de la base, se utiliza la actualización continua. En este caso, cada actualización de los datos en la réplica genera una sincronización con el repositorio principal en tiempo real.

Forma, Flecha

Descripción generada automáticamente

Al utilizar la técnica de fragmentación, el sistema divide la relación original o tabla en varios fragmentos y guarda cada fragmento en un sitio diferente.

Deben respetarse tres reglas:

1-la integridad, que impide la pérdida de información; es decir, si una relación o tabla es descompuesta en fragmentos, cada dato de la tabla original debe encontrarse en uno de los fragmentos.

2-La reconstrucción, que implica preservar dependencias, debe siempre permitir reconstruir la relación global a partir de los fragmentos.

3-Para entender la tercera regla es necesario conocer que hay tres esquemas de fragmentación de las relaciones:

* fragmentación horizontal,
* vertical
* mixta, que veremos en detalle más adelante.

4-El desacoplamiento, que es otra de las reglas necesarias de la fragmentación, establece que si la relación original es descompuesta horizontalmente en fragmentos, un mismo dato no puede estar en dos fragmentos diferentes. Si la relación original es descompuesta verticalmente, su clave primaria debe estar en todos los fragmentos.

Tabla

Descripción generada automáticamente

La fragmentación horizontal se usa comúnmente para situaciones en que las ubicaciones específicas generalmente solo necesitan acceso a la base de datos para gestionar los registros correspondientes a su dominio.

Veamos un ejemplo a partir de una tabla original *Clientes*, donde sus atributos son **el código**, el **nombre,** la **categoría de compra** (mayorista o minorista), la **sucursal** donde compra y **el saldo actual de la deuda**. Hay seis registros en este ejemplo.

Aplicando fragmentación horizontal, la relación *R* (la tabla original *Clientes*) se divide en varios subconjuntos. Cada registro de la relación original debe pertenecer como mínimo a uno de los fragmentos de modo que se pueda reconstruir la relación original si fuera necesaria.

En el ejemplo, tenemos dos fragmentos: se separaron los clientes minoristas y los mayoristas. También podría haberse realizado una fragmentación de acuerdo a la sucursal, resultando en tres fragmentos: clientes minoristas de la sucursal Tigre, clientes minoristas de la sucursal Belgrano, y clientes minoristas de la sucursal Quilmes.

Con la fragmentación vertical, la tabla original se fragmenta en función de los atributos. Tenemos entonces los fragmentos *Clientes por sucursal* y *Clientes deuda*. Este tipo de formato funciona bien para situaciones en las que las sedes de la base de datos distribuida interactúan con las mismas cuentas, pero orientadas a diferentes dominios, como por ejemplo la información de contacto del cliente y consulta de cuestiones financieras.

En la fragmentación mixta, se aplican ambas fragmentaciones en forma secuencial, siendo indistinto el orden. En nuestro ejemplo, la tabla original *Clientes* es fragmentada horizontalmente y obtenemos el fragmento *Clientes minoristas*. A partir de este fragmento, se aplica fragmentación vertical, resultando en *Clientes minoristas por sucursal* y *Clientes minoristas deuda*. En todos los casos, se conserva la clave primaria para poder acceder a los datos de los fragmentos y para reconstruir la tabla original si fuera necesario.

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamenteTabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para asegurar las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) de las transacciones que se ejecutan en un nodo, el sistema gestor de la base de datos distribuida contiene dos subsistemas: el sistema coordinador de transacciones y el gestor de transacciones.

1-El sistema coordinador de transacciones coordina la ejecución de las diferentes transacciones, tanto locales como globales, iniciadas en un nodo. Comienza la ejecución de una transacción descomponiéndola en un número de subtransacciones y las distribuye en los sitios apropiados para su ejecución. Coordina la terminación de una transacción, lo que puede resultar en que las transacciones estén comprometidas en todos los sitios o abortadas en todos ellos. Es el encargado de la ejecución del protocolo de compromiso en dos fases, conocido como "handshaking and two-phase commit".

La atomicidad de las transacciones es un aspecto crítico de la construcción de un sistema distribuido de base de datos para evitar un estado de inconsistencia. El protocolo de compromiso de dos fases asegura que tales situaciones no se produzcan. La idea básica de este protocolo es que el coordinador comprometa la transacción solo si la transacción alcanza el estado preparada en cada sitio donde debería ejecutarse. En caso contrario, por ejemplo, si la transacción se canceló en algún sitio, el sistema coordinador cancela la transacción en todos los sitios donde se estaba ejecutando. Todos los sitios donde la transacción se ejecutó deben acatar la decisión del sistema coordinador.

Si un sitio falla mientras una transacción se está ejecutando, cuando el sitio se recupere del fallo, debería estar en posición de cancelar la transacción y volver todos los registros al estado anterior.

**El control de concurrencia** es otra característica de una base de datos distribuida. Como una transacción puede acceder a elementos de datos de varios sitios, los administradores de transacciones de esos sitios pueden necesitar coordinarse para implementar el control de concurrencia. Si **se utiliza el bloqueo**, como sucede habitualmente, este se puede realizar de forma local en los sitios que contienen los elementos de datos accedidos. Sin embargo, también existe la posibilidad de un interbloqueo que involucre las transacciones originadas en múltiples sitios.

2-El gestor de transacciones gestiona la ejecución de las transacciones o subtransacciones que acceden a los datos almacenados en el sitio local. Estas transacciones pueden ser locales o formar parte de una transacción global. Es el encargado de definir la estructura de las transacciones, mantener la consistencia en la base de datos cuando se ejecuta una transacción o se cancela su ejecución, mantener protocolos de fiabilidad, implementar algoritmos para el control de la concurrencia y sincronizar las transacciones que se ejecuten simultáneamente. El sistema gestor recibe solicitudes de procesamiento de transacciones del sistema coordinador y las traduce en acciones.

Diagrama

Descripción generada automáticamente

Para propagar cualquier modificación que se haga sobre un registro a todas las copias almacenadas de ese registro se cuenta con dos estrategias: la estrategia de propagación inmediata desde la sede primaria donde se produjo la actualización a todas las réplicas en tiempo real y la propagación diferida, en la que la sede primaria se encarga de actualizar la siguiente copia y esta a la siguiente en forma secuencial con posterioridad a la modificación.