Bases de Datos Distribuidas

Vicente Toledo – Israel Miralles

Indice

1.-¿Que son Bases de Datos Distribuidas?

-Son un grupo de datos que pertenecen a un sistema pero a su vez esta reapartido entre ordenadores de una misma red, ya sea a nivel local o cada uno en una diferente localizacion geografica, cada sitio en la red es autónomo en sus capacidades de procesamiento y es capaz de realizar operaciones locales y en cada uno de estos ordenadores debe estar ejecutandose una aplicación a nivel global que permita la consulta de todos los datos como si se tratase de uno solo.

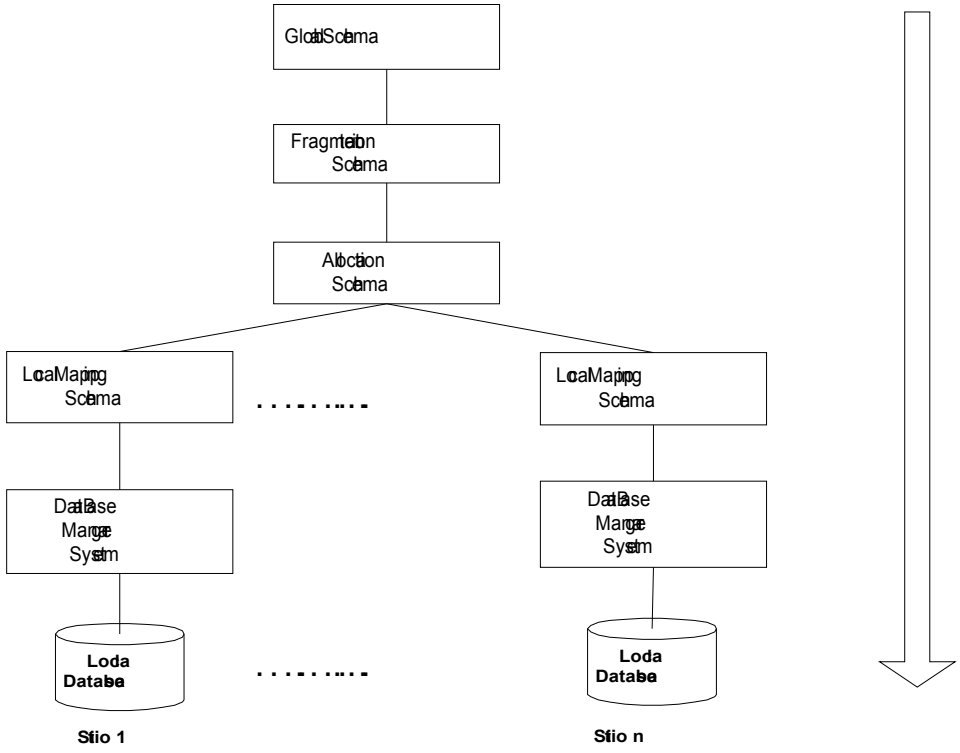
1.- Comparación

Centralizado	**Distribuido**
Control centralizado: un solo DBA	Control jerárquico: DBA global y DBA local
Independencia de Datos: Organización de los datos es transparente para el programador	Transparencia en la Distribución: Localización de los datos es un aspecto adicional de independencia de datos
Reducción de redundancia: Una sola copia de datos que se comparta	Replicación de Datos: Copias múltiples de datos que incrementa la localidad y la disponibilidad de datos
Estructuras físicas complejas para accesos eficientes	No hay estructuras intersitios. Uso de optimización global para reducir transferencia de datos
Seguridad	Problemas de seguridad intrínsecos

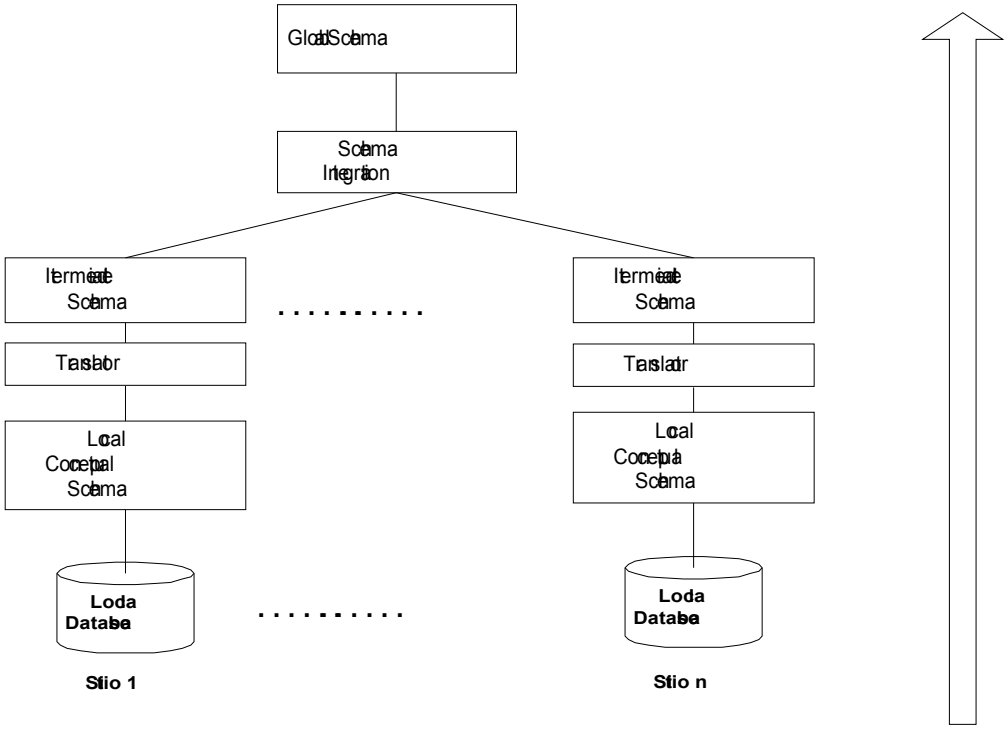
-Para tener una base de datos distribuida debe cumplirse las condiciones de una Red Computacional. Una red de comunicación provee las capacicades para que un proceso ejecutandose en un sitio de la red envie y reciba mensajes de otro proceso ejecutandose en un sitio distinto. Parámetros a considerar incluyen: Retraso en la entrega de mensajes, Costo de transmisión de un mensaje y Confiabilidade de la red. Diferentes tipos de redes: point-to-point, broadcast, lan, wan.

2.-Arquitectura de las Bases de Datos

Integración lógica por medio de diseño top-down (DistDB)



Integración lógica por medio de bottom-up (Multidatabase)



-Global Schema: Define todos los datos que están incluidos en la bd distribuida tal como si la bd no fuera distribuida. Consiste de una definición de relaciones globales.

-Fragmentation Schema: Traducción entre relaciones globales y fragmentos. (Una relación global puede consistir de varios *fragmentos* pero un fragmento está asociado con sólo una relación global)

-Allocation Schema: Define el sitio (o sitios) en el cual un fragmennto está localizado.

-Local Maping Schema: Traduce los fragamentos locales a los objetos que son manejados por el SMBD local

Separación entre fragmentación y localización.

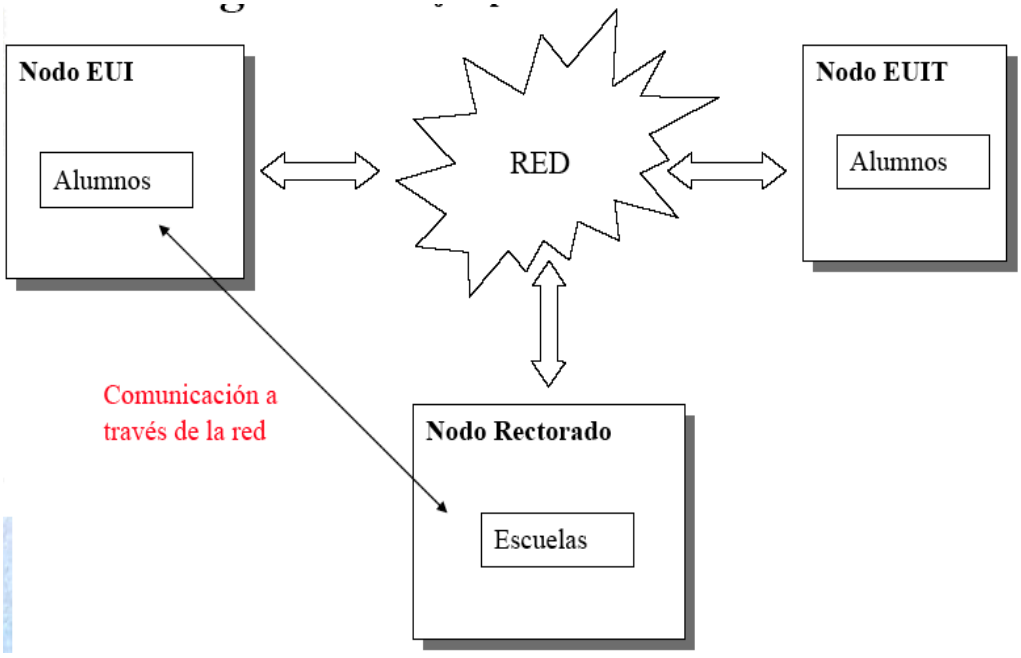
-Transparencia de Fragmentación

-Transparencia de Localización

-Control explícito de redundancia

-Independencia de BD locales

1.-Ejemplo de una Base de Datos Distribuidas



- **Nodos de las Escuelas:**

DNI	Escuela	Nombre	Nota ingreso	Beca

- **Nodo del Rectorado:**

Escuela	Situación	Número alumnos

- **Nuevo alumno en la secretaría del centro: transacción local.**
- **Nuevo alumno en el rectorado: transacción global**

3.-Tipos de almacenamiento

1-Replica

El sistema conserva varias copias o réplicas idénticas de unatabla. Cada réplica se almacena en un nodo diferente.

Ventajas:

Disponibilidad: El sistema sigue funcionando aún en caso de caída de uno de los nodos.

Aumento del paralelismo: Varios nodos pueden realizar consultas en paralelo sobre la misma tabla. Cuantas más réplicas existan de la tabla, mayor serála posibilidad de que el dato buscado se encuentre en el nodo desde el que se realiza la consulta, minimizando con ello el tráfico de datos entre nodos.

Inconveniente:

Aumento de la sobrecarga en las actualizaciones: El sistema debe asegurar que todas las réplicas de la tabla sean consistentes. Cuandose realiza una actualización sobre una de las réplicas, los cambios deben propagarse a todas las réplicas de dicha tabla a lo largo del sistema distribuido.

2.-Fragmentación

Existen tres tipos de fragmentacion la horizontal, la vertical y la mixta

1.-Fragmentación Horizontal

Unatabla T se divide en subconjuntos, T1, T2, ...Tn. Los fragmentos se definen a través de una operación de selección y su reconstrucción se realizaráconuna operación de unión de los fragmentos componentes.

Cada fragmento se sitúa en un nodo.

Pueden existir fragmentos no disjuntos: combinación de fragmentación y replicación.

Ejemplo:

Tabla inicial de alumnos

DNI	Escuela	Nombre	Nota ingreso	Beca
87633483	EUI	Concha Queta	5.6	No
99855743	EUI	Josechu Letón	7.2	Si
33887293	EUIT	Oscar Romato	6.1	Si
05399075	EUI	Bill Gates	5.0	No
44343234	EUIT	Pepe Pótamo	8.0	No
44543324	EUI	Maite Clado	7.5	Si
66553234	EUIT	Ernesto Mate	6.6	No

Tabla de alumnos fragmentada

Fragmento de la EUI: $\sigma_{Escuela="EUI"}(T)$

DNI	Escuela	Nombre	Nota ingreso	Beca
87633483	EUI	Concha Queta	5.6	No
99855743	EUI	Josechu Letón	7.2	Si
05399075	EUI	Bill Gates	5.0	No
44543324	EUI	Maite Clado	7.5	Si

Fragmento de la EUIT: $\sigma_{Escuela="EUIT"}(T)$

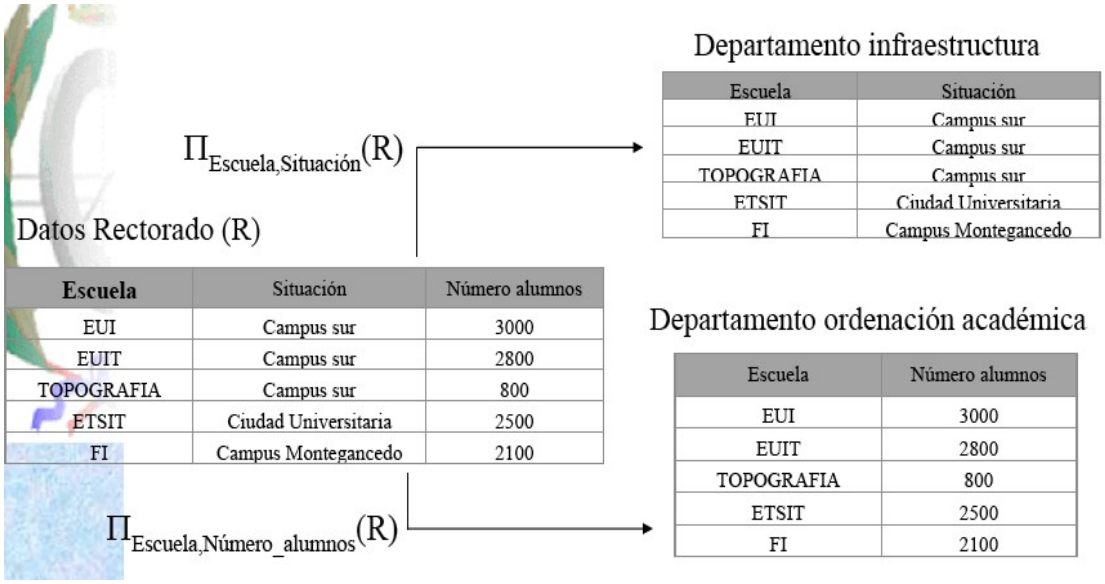
DNI	Escuela	Nombre	Nota ingreso	Beca
33887293	EUIT	Oscar Romato	6.1	Si
44343234	EUIT	Pepe Pótamo	8.0	No
66553234	EUIT	Ernesto Mate	6.6	No

2.-Fragmentación Vertical

Una tabla T se divide en subconjuntos, T1, T2, ...Tn. Los fragmentos se definen a través de una operación de proyección.

Cada fragmento debe incluir la clave primaria de la tabla. Su reconstrucción se realizará con una operación de join de los fragmentos componentes, pueden existir fragmentos no disjuntos: combinación de fragmentación y replicación.

Ejemplo:



Escuela	Situación
EUI	Campus sur
EUIT	Campus sur
TOPOGRAFIA	Campus sur
ETSIT	Ciudad Universitaria
FI	Campus Montegancedo

Escuela	Situación	Número alumnos
EUI	Campus sur	3000
EUIT	Campus sur	2800
TOPOGRAFIA	Campus sur	800
ETSIT	Ciudad Universitaria	2500
FI	Campus Montegancedo	2100

Escuela	Número alumnos
EUI	3000
EUIT	2800
TOPOGRAFIA	800
ETSIT	2500
FI	2100

3.-Fragmentación Mixta

Como el mismo nombre indica es una combinación de las dos anteriores vistas he aquí un ejemplo apartir de una tabla fragmentada horizontalmente.

DNI	Escuela	Nombre	Beca
87633483	EUI	Concha Queta	No
99855743	EUI	Josechu Letón	Si
05399075	EUI	Bill Gates	No
44543324	EUI	Maite Clado	Si

$\Pi_{DNI,Escuela,Escuela,Nombre,Beca}(E)$

Fragmento de la EUI: $\sigma_{Escuela="EUI"}(T)$

DNI	Escuela	Nombre	Nota ingreso	Beca
87633483	EUI	Concha Queta	5.6	No
99855743	EUI	Josechu Letón	7.2	Si
05399075	EUI	Bill Gates	5.0	No
44543324	EUI	Maite Clado	7.5	Si

$\Pi_{DNI,Escuela,Nombre,Nota\ ingreso}(E)$

DNI	Escuela	Nombre	Nota ingreso
87633483	EUI	Concha Queta	5.6
99855743	EUI	Josechu Letón	7.2
05399075	EUI	Bill Gates	5.0
44543324	EUI	Maite Clado	7.5

3.-Replica y Fragmentación

Las técnicas de réplica y fragmentación se pueden aplicar sucesivamente a la misma relación de partida. Un fragmento se puede replicar y a su vez esa réplica ser fragmentada, para luego replicar alguno de esos fragmentos.

4.-Niveles de Transparencia en una Base de Datos Distribuida

El propósito de establecer una arquitectura de un sistema de bases de datos distribuidas es ofrecer un nivel de transparencia adecuado para el manejo de la información.

La transparencia se define como la separación de la semántica de alto nivel de un sistema de los aspectos de bajo nivel relacionados a la implementación del mismo. Un nivel de transparencia adecuado permite ocultar los detalles de implementación a las capas de alto nivel de un sistema y a otros usuarios.

El sistema de bases de datos distribuido permite proporcionar independencia de los datos.

La independencia de datos se puede dar en dos aspectos: lógica y física.

.1 **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. Esto permite que un cambio en la definición de un esquema no debe afectar a las aplicaciones de usuario. Por ejemplo, el agregar un nuevo atributo a una relación, la creación de una nueva relación, el reordenamiento lógico de algunos atributos.

.2 **Independencia física de 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 de usuario. Por ejemplo, los datos pueden ser movidos de un disco a otro, o la organización de los datos puede cambiar.

La transparencia al nivel de red se refiere a que los datos en un SBDD se accedan sobre una red de computadoras, sin embargo, las aplicaciones no deben notar su existencia. La transparencia al nivel de red conlleva a dos cosas:

.1Transparencia sobre la localización de datos. el comando que se usa es independiente de la ubicación de los datos en la red y del lugar en donde la operación se lleve a cabo. Por ejemplo, en Unix existen dos comandos para hacer

una copia de archivo. Cp se utiliza para copias locales y rcp se utiliza para copias remotas. En este caso no existe transparencia sobre la localización.

.2Transparencia sobre el esquema de nombramiento. Lo anterior se logra proporcionando un nombre único a cada objeto en el sistema distribuido. Así, no se debe mezclar la información de la localización con en el nombre de un objeto.

La transparencia sobre replicación de datos se refiere a que si existen réplicas de objetos de la base de datos, su existencia debe ser controlada por el sistema no por el usuario. Se debe tener en cuenta que cuando el usuario se encarga de manejar las réplicas en un sistema, el trabajo de éste es mínimo por lo que se puede obtener una eficiencia mayor. Sin embargo, el usuario puede olvidarse de mantener la consistencia de las réplicas teniendo así datos diferentes.

La transparencia a nivel de fragmentación de datos permite que cuando los objetos de la bases de datos están fragmentados, el sistema tiene que manejar la conversión de consultas de usuario definidas sobre relaciones globales a consultas definidas sobre fragmentos. Así también, será necesario mezclar las respuestas a consultas fragmentadas para obtener una sola respuesta a una consulta global. El acceso a una base de datos distribuida debe hacerse en forma transparente.

En resumen, la transparencia tiene como punto central la independencia de datos.

La responsabilidad sobre el manejo de transparencia debe estar compartida tanto por el sistema operativo, el sistema de manejo de bases de datos y el lenguaje de acceso a la base de datos distribuida. Entre estos tres módulos se deben resolver los aspectos sobre el procesamiento distribuido de consultas y sobre el manejo de nombres de objetos distribuidos.

5.-Procesamiento Distribuido de Consultas

El procesamiento de consultas es de suma importancia en bases de datos centralizadas. Sin embargo, en BDD éste adquiere una relevancia mayor. El objetivo es convertir transacciones de usuario en instrucciones para manipulación de datos. No obstante, el orden en que se realizan las transacciones afecta grandemente la velocidad de respuesta del sistema. Así, el procesamiento de consultas presenta un problema de optimización en el cual se determina el orden en el cual se hace la menor cantidad de operaciones. En BDD se tiene que considerar el procesamiento local de una consulta junto con el costo de transmisión de información al lugar en donde se solicitó la consulta.

6.-Recuperación

En los entornos distribuidos de datos podemos encontrar lo siguientes:

Fallo de los nodos. Cuando un nodo falla, el sistema deberá continuar trabajando con los nodos que aún funcionan. Si el nodo a recuperar es una base de datos local, se deberán separar los datos entre los nodos restantes antes de volver a unir de nuevo el sistema.

Copias múltiples de fragmentos de datos. El subsistema encargado del control de concurrencia es el responsable de mantener la consistencia en todas las copias que se realicen y el subsistema que realiza la recuperación es el responsable de hacer copias consistentes de los datos de los nodos que han fallado y que después se recuperarán.

Transacción distribuida correcta. Se pueden producir fallos durante la ejecución de una transacción correcta si se plantea el caso de que al acceder a alguno de los nodos que intervienen en la transacción, dicho nodo falla.

Fallo de las conexiones de comunicaciones. El sistema debe ser capaz de tratar los posibles fallos que se produzcan en las comunicaciones entre nodos. El caso mas extremo es el que se produce cuando se divide la red. Esto puede producir la separación de dos o más particiones donde las particiones de cada nodo pueden comunicarse entre si pero no con particiones de otros nodos.
Para implementar las soluciones a estos problemas, supondremos que los datos se encuentran almacenados en un único nodo sin repetición. De ésta manera sólo existirá un único catálogo y un único DM (Data Manager) encargados del control y acceso a las distintas partes de los datos.
Para mantener la consistencia de los datos en el entorno distribuido contaremos con los siguientes elementos:

Catálogo: Programa o conjunto de programas encargados de controlar la ejecución concurrente de las transacciones.

CM (Cache Manager). Subsistema que se encarga de mover los datos entre las memorias volátiles y no volátiles, en respuesta a las peticiones de los niveles más altos del sistema de bases de datos. Sus operaciones son Fetch(x) y Flush(x).

RM (Recovery Manager). Subsistema que asegura que la base de datos contenga los efectos de la ejecución de transacciones correctas y ninguno de incorrectas. Sus operaciones son Start, Commit, Abort, Read, Write, que utilizan a su vez los servicios del CM.

DM (Data Manager). Unifica las llamadas a los servicios del CM y el RM.

TM (Transaction Manager). Subsistema encargado de determinar que nodo deberá realizar cada operación a lo largo de una transacción.

Las operaciones de transacción que soporta una base de datos son: Start, Commit y Abort. Para comenzar una nueva transacción se utiliza la operación Start. Si aparece una operación commit, el sistema de gestión da por terminada la transacción con normalidad y sus efectos permanecen en la base de datos. Si, por el contrario, aparece una operación abort, el sistema de gestión asume que la transacción no termina de forma normal y todas las modificaciones realizadas en la base de datos por la transacción deben de ser deshechas.

7.-Ventajas y Desventajas

1.-Ventajas

Los sistemas de bases de datos distribuidos tienen múltiples ventajas. En primer lugar los datos son localizados en lugar más cercano, por tanto, el acceso es más rápido, el procesamiento es rápido debido a que varios nodos intervienen en el procesamiento de una carga de trabajo, nuevos nodos se pueden agregar fácil y rápidamente. La comunicación entre nodos se mejora, los costos de operación se reducen, son amigables al usuario, la probabilidad de que una falla en un solo nodo afecte al sistema es baja y existe una autonomía e independencia entre los nodos.

Las razones por las que compañías y negocios migran hacia bases de datos distribuidas incluyen razones organizacionales y económicas, para obtener una interconexión confiable y flexible con las bases de datos existentes, y por un crecimiento futuro. El enfoque distribuido de las bases de datos se adapta más naturalmente a la estructura de las organizaciones. Además, la necesidad de desarrollar una aplicación global (que incluya a toda la organización), se resuelva fácilmente con bases de datos distribuidas. Si una organización crece por medio de la creación de unidades o departamentos nuevos, entonces, el enfoque de bases de datos distribuidas permite un crecimiento suave.

Los datos se pueden colocar físicamente en el lugar donde se accedan más frecuentemente, haciendo que los usuarios tengan control local de los datos con los que interactúan. Esto resulta en una autonomía local de datos permitiendo a los usuarios aplicar políticas locales respecto del tipo de accesos a sus datos.

Mediante la replicación de información, las bases de datos distribuidas pueden presentar cierto grado de tolerancia a fallos haciendo que el funcionamiento del sistema no dependa de un solo lugar como en el caso de las bases de datos centralizadas.

La independencia de datos se puede dar en dos aspectos: lógica y física.

2.-Desventajas

Los sistemas de bases de datos distribuidos tienen múltiples ventajas. En primer lugar los datos son localizados en lugar más cercano, por tanto, el acceso es más rápido, el procesamiento es rápido debido a que varios nodos intervienen en el procesamiento de una carga de trabajo, nuevos nodos se pueden agregar fácil y rápidamente. La comunicación entre nodos se mejora, los costos de operación se reducen, son amigables al usuario, la probabilidad de que una falla en un solo nodo afecte al sistema es baja y existe una autonomía e independencia entre los nodos.

Las razones por las que compañías y negocios migran hacia bases de datos distribuidas incluyen razones organizacionales y económicas, para obtener una interconexión confiable y flexible con las bases de datos existentes, y por un crecimiento futuro. El enfoque distribuido de las bases de datos se adapta más naturalmente a la estructura de las organizaciones. Además, la necesidad de desarrollar una aplicación global (que incluya a toda la organización), se resuelva fácilmente con bases de datos distribuidas. Si una organización crece por medio de la creación de unidades o departamentos nuevos, entonces, el enfoque de bases de datos distribuidas permite un crecimiento suave.

Los datos se pueden colocar físicamente en el lugar donde se accedan más frecuentemente, haciendo que los usuarios tengan control local de los datos con los que interactúan. Esto resulta en una autonomía local de datos permitiendo a los usuarios aplicar políticas locales respecto del tipo de accesos a sus datos.

Mediante la replicación de información, las bases de datos distribuidas pueden presentar cierto grado de tolerancia a fallos haciendo que el funcionamiento del sistema no dependa de un solo lugar como en el caso de las bases de datos centralizadas.

La independencia de datos se puede dar en dos aspectos: lógica y física.

8.-Bibliografía

http://usuarios.lycos.es/jrodr35/

http://html.rincondelvago.com/bases-de-datos-distribuidas_1.html

http://sacbeob.8m.com/tutoriales/bddistribuidas/index.htm

http://www.cs.cinvestav.mx/SC/prof_personal/adiaz/Disdb/Cap_1.html