

Sistemas Distribuidos

Bibliografía: Introducción a los Sistemas de Bases de Datos
Date, C.J.

Bases de datos distribuidas

- implica que una sola aplicación deberá ser capaz de trabajar en forma “**transparente**”
- con datos dispersos en varias BD diferentes,
- administradas por varios DBMS distintos,
- ejecutadas en varias máquinas diferentes,
- apoyadas por diversos sistemas operativos
- y conectadas entre sí mediante varias redes de comunicación distintas.

Bases de datos distribuidas

“Transparente” significa
que la aplicación trabajaría,
desde un **punto de vista lógico**,
como si **un solo DBMS**,
ejecutado en **una sola máquina**,
administrara **todos los datos**

Sistema de bases de datos distribuidas (BDD)

- se compone de un conjunto de **sitios** conectados entre sí mediante algún tipo de red de comunicaciones, en el cual
 - **Cada sitio es un sistema de BD en sí mismo.**
 - Es decir, cada sitio tiene sus **BD reales locales**, sus propios **usuarios locales**, sus propios **DBMS** y **programas para la administración de transacciones** (incluyendo sus propios programas de bloqueo, bitácoras, recuperación, etc.)

Sistema de bases de datos distribuidas (BDD)

- Pero **los sitios han convenido en trabajar juntos** (si es necesario) **con el fin** de que:
 - un usuario de cualquier sitio pueda obtener acceso a los datos de cualquier punto de la red tal como si **todos los datos estuvieran almacenados en el sitio propio del usuario.**

- Un sistema de **BDD** puede considerarse como una especie de **sociedad entre los DBMS individuales locales de todos los sitios.**
- Un **nuevo componente de software** en cada sitio realiza las funciones de **sociedad necesarias**

Es la combinación de
este **nuevo componente**
y el **DBMS** ya existente
lo que constituye el llamado
Sistema de Administración de bases de
datos distribuidas: **DDBMS**.

Ventajas

- ¿Por qué son deseables las bases de datos distribuidas?
- Por lo regular las **empresas ya están distribuidas**, por lo menos desde el **punto de vista lógico** (en divisiones, departamentos, etc.) **y en el sentido físico** (plantas, talleres, laboratorios, etc.)
- Por lo tanto la **información está distribuida también**

Ventajas

- Un sistema distribuido permite que la **estructura de la BD refleje la estructura de la empresa:**
 - Los **datos locales** se pueden mantener **en forma local**, donde por lógica deben estar, pero al mismo tiempo
 - Es posible **obtener acceso a datos remotos** en caso necesario.

Ejemplo

- Supongamos un **sistema bancario y dos sitios**: Rosario y Bs.As.:
 - Los registros de las cuentas de Rosario están en Rosario y las de Bs.As. en Bs.As.
 - Esto da **eficiencia** al procesamiento:
 - los datos están en el lugar donde se los utiliza con mayor frecuencia.
 - es posible tener acceso a una cuenta de Rosario desde Bs.As. y viceversa.

Desventajas

- **Falta de experiencia** generalizada (pocas aplicaciones: reservas aéreas)
- Si no hay un buen diseño y organización trae **mayor complejidad**:
- problemas del centralizado + problemas del distribuido
- Puede **aumentar costos iniciales**: Hardware y software de comunicación y distribución
- **Seguridad**: se debe **aumentar los controles** respecto a BD centralizadas
- **Complejidad** de los sistemas distribuidos (desde el **punto de vista técnico**)

El principio fundamental de las Bases de Datos Distribuidas (regla 0)

- Desde el punto de vista del usuario, un sistema distribuido deberá ser idéntico a un sistema no distribuido.

Esto es:

- Las operaciones de **DML** no deberán **sufrir cambios**.
- Las operaciones de **DDL** requerirán **cierta ampliación**.
 - Ejemplo: poder **crear** una tabla en el **sitio X** y poder **almacenarla** en el **sitio Y**.

Las doce reglas

1. Autonomía local
2. No dependencia de un sitio central
3. Operación continua
4. Independencia con respecto a la localización
5. Independencia con respecto a la fragmentación
6. Independencia de réplica

Las doce reglas

7. Procesamiento distribuido de consultas
8. Manejo distribuido de transacciones
9. Independencia con respecto al equipo
10. Independencia con respecto al sistema operativo
11. Independencia con respecto a la red
12. Independencia con respecto al DBMS

1. Autonomía local

- Los **sitios** deben ser **autónomos**.
- Todas las **operaciones** en un sitio dado se **controlan** de ese **sitio**.
- **Ningún sitio X** debe **depender** de **otro** sitio Y para su correcto funcionamiento.
- Si **cae Y**, X debe poder **seguir** **trabajando**.

2. No dependencia de un sitio central

- Todos los **sitios** deben tratarse **por igual**.
- **No** debe haber **dependencia** de un **sitio central** para obtener un servicio, por ejemplo procesar una consulta.
- Si el sitio central sufriera un desperfecto todo el sistema dejaría de funcionar

3. Operación continua

- **Nunca** debería haber necesidad de **apagar el sistema** para realizar alguna función.
- Por ejemplo para:
 - añadir un nuevo sitio o
 - instalar una versión del DBMS existente.

4. Independencia con respecto a la localización

- Los usuarios no deberían necesitar saber **dónde** están **almacenados** físicamente los **datos**.
- Debe **comportarse** desde el punto de vista **lógico** como si todos los **datos** estuvieran **almacenados** en su **propio sitio local**.

5. Independencia con respecto a la fragmentación

- Un sistema maneja ***fragmentación de los datos*** si es posible **dividir una relación** en partes o “fragmentos” para propósitos de **almacenamiento físico**.

5. Independencia con respecto a la fragmentación

- Los **datos** pueden **almacenarse** en el lugar donde se los utiliza con **más frecuencia**:
 - la **mayor** parte de las **operaciones** serán **locales** y
 - se **reducirá** el **tráfico** en la red.
 - **Ejemplo**:
 - empleados de Rosario en Rosario,
 - los de Buenos Aires en Buenos Aires.

5. Independencia con respecto a la fragmentación

- Existen dos tipos de **fragmentación**:
 - **horizontal**
 - **vertical**
- se corresponden con la
 - **selección**
 - **proyección**

5. Independencia con respecto a la fragmentación

- En la **proyección** deben **conservar la clave primaria**
- La **reconstrucción** de los fragmentos se hace **mediante reunión y unión**
 - **Reunión** en caso de fragmentos **verticales**.
 - **Unión** en caso de fragmentos **horizontales**.

5. Independencia con respecto a la fragmentación

- **Independencia con respecto a la fragmentación** significa:
 - Los **usuarios** tendrán una **vista de los datos** con fragmentos combinados lógicamente mediante reuniones y uniones apropiadas.
 - El **optimizador** determina a cuáles **fragmentos físicos** es necesario **tener acceso** para satisfacer cualquier solicitud del usuario.

6. Independencia de réplica

- Un sistema maneja *independencia de réplica de datos* si una **relación** dada (o un **fragmento**) se puede representar en el nivel físico mediante **varias copias** almacenadas o réplicas, **en muchos sitios** distintos.

6. Independencia de réplica

- **Independencia de réplica** significa que los usuarios deberán comportarse lógicamente como si existiera **una sola copia**.
- Debe ser **transparente** para el usuario.

6. Independencia de réplica

- **La réplica es deseable por dos razones:**
 - Puede producir un **mejor desempeño**: las **aplicaciones** pueden **operar sobre copias locales** en vez de tener que comunicarse con sitios remotos
 - **Mejor disponibilidad**: un **objeto está disponible** para su procesamiento en tanto esté disponible **por lo menos una copia**, al menos para propósitos de recuperación.

6. Independencia de réplica

- **Desventaja principal:**
 - Problema de **propagación de actualizaciones**
 - al actualizar un cierto objeto, deben **actualizarse todas sus réplicas**

7. Procesamiento distribuido de consultas

- Hay una **optimización global** más una **optimización local en cada sitio**.
- La **optimización** es más importante en un **sistema distribuido** que en el centralizado.
- Hay muchas maneras de **trasladar datos entre varios sitios**.
- Hay que **encontrar la estrategia más eficiente**.

7. Procesamiento distribuido de consultas

Por **ejemplo**:

- una **solicitud de unión** de una relación Rx almacenada en el sitio X y una Ry en Y, podría llevarse a cabo:
 - trasladando Rx a Y ó
 - trasladando Ry a X ó
 - trasladando las dos a un tercer sitio.
- **Según la estrategia, el tiempo puede variar** entre un segundo y dos días.

8. Manejo distribuido de transacciones

El manejo de transacciones **comprende:**

- **Control de recuperación**
 - Una **transacción** debe ser **atómica** (todo o nada)
 - En las BDD el sistema debe asegurarse que **todos los agentes** correspondientes a la transacción **se comprometan o retrocedan al unísono.**
 - Esto se logra mediante un **protocolo de compromiso de dos fases.**

8. Manejo distribuido de transacciones

El manejo de transacciones **comprende:**

- **Control de concurrencia**
 - Esta basado en el **bloqueo**, igual que en sistemas no distribuidos.

9. Independencia con respecto al equipo

- Es conveniente ejecutar el **mismo DBMS en diferentes equipos** y presentar al usuario **una sola imagen del sistema.**

10. Independencia con respecto al sistema operativo

- Se debe poder **ejecutar el mismo DBMS en diferentes equipos y sistemas operativos.**

11. Independencia con respecto a la red

- Se debe poder **manejar varias redes de comunicación distintas.**

12. Independencia con respecto al DBMS

- Los DBMS en los distintos sitios deben **manejar la misma interfaz.**
- **Por ejemplo:**
 - si tanto INGRES como Oracle manejan la norma oficial de SQL,
 - es posible una comunicación entre los dos en el contexto de un sistema distribuido.

Conclusiones

- **No todas** las reglas serán **pertinentes** en **todas las situaciones**
- **No todas** las reglas son **independientes** entre sí
- **No todas** las reglas tienen la **misma importancia**
- Las reglas son **útiles** para **entender la tecnología distribuida**

Conclusiones

Un **objetivo** primordial
en los **sistemas distribuidos**
es **reducir al mínimo**
el número y volumen de los mensajes