



INSTITUTO POLITÉCNICO
NACIONAL



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS

PROTOCOLOS DE INTERNET 2TV12

Arquitecturas de Bases de Datos Distribuidas

Autores:

Bernal Servin Alonso
Roa Jimenez Eduardo Yared
Zamora Florez Rafael

Profesor:

De la Cruz Sosa Carlos

Reporte de lectura 01

09 de Marzo de 2022

Índice

| | |
|---|---|
| 1. Introducción | 2 |
| 2. Arquitecturas distribuidas DBMS | 2 |
| 3. Arquitectura ANSI/SPARC | 3 |
| 4. Una arquitectura centralizada DBMS genérica | 3 |
| 5. Modelos arquitectónicos para DBMS distribuidos | 5 |
| 6. Autonomía | 5 |
| 7. Distribución | 6 |
| 8. Heterogeneidad | 6 |
| 9. Alternativas arquitectónicas | 6 |
| 10.Sistemas cliente/servidor | 7 |
| 11.Sistema peer to peer | 7 |
| 12.Arquitectura de un sistema de base de datos múltiple | 8 |
| Referencias | 9 |

1. Introducción

El proposito de este reporte es presentar los aspectos relevantes de las arquitecturas disribuidas de los sistemas de manejo de bases de datos (DBMS).

2. Arquitecturas distribuidas DBMS

Las D B M S distribuidos son el principal exponente de la distribución horizontal de toda la funcionalidad de la BD y de los datos particionados/replicados a los nodos de procesamiento conectados por una red. En consecuencia, los requisitos básicos se mantienen, lo que conduce a un modelo arquitectónico que consiste en modelos de capas idénticas para cada nodo, junto con una capa de conexión responsable de los servicios de comunicación, adaptación o mediación. Por el contrario, la distribución vertical se consigue normalmente mediante los llamados D B M S cliente/servidor. Su principal preocupación es hacer que la capacidad de procesamiento del D B M S esté disponible cerca de la aplicación en el cliente (ordenador). Por lo general, los D B M S cliente/servidor se utilizan en aplicaciones que dependen de transacciones de larga duración con un mecanismo de checkout/checkin de datos (versionados). Sin embargo, en todos estos casos, el objetivo principal de la arquitectura por capas es el procesamiento orientado a la navegación o a los conjuntos de datos, como se indica en el cuadro.

Table 1: DBMS mapping hierarchy

| | Level of abstraction | Objects |
|----|----------------------|---------------------------------|
| L5 | Nonprocedural access | Tables, views, tuples |
| L4 | Navigational access | Logical records, sets, networks |
| L3 | Access path mgmt | Physical records, access paths |
| L2 | Propagation control | Segments, pages |
| L1 | File mgmt | Files, blocks |

Figura 1: Jerarquía de mapeo de una DBMS

Hemos observado que las invariantes en la gestión de bases de datos determinan los pasos de asignación de la arquitectura de apoyo y, por lo tanto, un modelo arquitectónico sirve bien siempre que pueda apoyar efectivamente estas invariantes básicas. Para el uso adecuado de la arquitectura en capas, sus invariantes básicas deben ser ciertas: mapeo orientado a la página, mapeo al almacenamiento externo, gestión de datos orientados a registros, procesamiento de bases de datos orientadas a conjuntos. Sin embargo, hoy en día, cada arquitectura de DBSM se ve inundada por una oleada de requisitos de nuevos tipos de datos, conceptos transaccionales y escenarios de gestión de datos. Y escenarios de gestión de datos en los que estas invariantes sólo se mantienen parcialmente o en los que tienen que ser sustituidas por conceptos totalmente nuevos. Hoy en día, los requisitos importantes de los SGBD incluyen flujos de datos, documentos no estructurados o semiestructurados, series temporales, objetos espaciales, etc. Debido a la variedad de estas demandas y tendencias de aplicación emergentes y di-

vergentes, debemos explorar seriamente la cuestión de si el potencial evolutivo

La arquitectura ANSI-SPARC es un estándar de alto nivel para el diseño de sistemas de gestión de bases de datos. ANSI-SPARC se caracteriza por su diseño en tres capas: externa, conceptual e interna. Esta arquitectura está diseñada para proporcionar una abstracción útil para simplificar el acceso a la base de datos en diferentes niveles de complejidad de los requisitos.

3. Arquitectura ANSI/SPARC

La norma ANSI-SPARC reconocía que existían dos niveles principales de estructura de bases de datos comunes entre las aplicaciones modernas (de la época). Estos niveles, el interno y el externo, distinguían la forma en que los datos eran vistos por el programador frente al desarrollador. La norma ANSI-SPARC esbozaba la necesidad de una tercera capa, la capa conceptual, utilizada para la "descripción de la empresa de la información tal y como se modela en la base de datos". Las tres capas de la arquitectura ANSI-SPARC se describen con más detalle así.

Capa externa

La vista del usuario de la base de datos contiene representaciones de datos que pueden representar abstracciones de mayor nivel que la forma en que los datos se almacenan realmente en la base de datos. Un ejemplo es una aplicación que produce gráficos dinámicos basados en datos numéricos de la base de datos. El usuario vería ilustraciones pero los datos reales serían números.

Capa conceptual

La capa conceptual define cómo se almacenan los datos dentro de toda la base de datos y proporciona una visión global de los mismos. Aquí se definen las relaciones entre los datos y los modelos conceptuales de los datos de la base de datos. Esta capa, aunque de menor nivel que la capa externa, sigue siendo independiente del hardware del software.

Capa interna

La capa interna define cómo se representa la base de datos, físicamente, en el sistema. Esto incluye la partición, la indexación y las adaptaciones relacionadas con el sistema operativo que puedan ser necesarias. Este nivel implica el menor grado de abstracción y a él acceden los desarrolladores de sistemas de gestión de bases de datos (SGBD), los desarrolladores de sistemas operativos, etc. El sistema de gestión de bases de datos centralizado es el sistema en el que todos los datos se almacenan y gestionan en una sola unidad. También se conoce como sistema de base de datos centralizado. Este sistema se utiliza sobre todo en una organización, en cualquier empresa o en una institución para centralizar las tareas. Se puede acceder a los datos a través de una red de área local (LAN) o de una red de área amplia (WAN). El ordenador central es un ejemplo de sistema de gestión de bases de datos centralizado.

4. Una arquitectura centralizada DBMS genérica

Funciones de la base de datos centralizada

- **Procesamiento de consultas distribuido:**

La función básica del sistema de gestión de bases de datos centralizadas es proporcionar facilidades y dar acceso a todos los ordenadores conectados que cumplen con todos los requisitos solicitados por cualquier nodo individual.

- **Unidad central única:**

Todos los datos y la información se almacenan en un único sistema de gestión de bases de datos centralizado. El sistema informático que cumple los requisitos de todos los ordenadores conectados se conoce como servidor y los demás ordenadores se conocen como clientes.

- **Transparencia:**

Todas las consultas se procesan en un único sistema informático también conocido como servidor. En este sistema de gestión no hay duplicación ni datos irrelevantes. Todos los ordenadores conectados tienen acceso al ordenador central para procesar sus consultas y necesidades.

- **Escalable:**

En este sistema de gestión de bases de datos centralizadas se pueden añadir varios ordenadores. Estos ordenadores están conectados al sistema a través de una red.

Ventaja del sistema de base de datos centralizada:

- **Integridad de los datos:**

Los datos están más unificados ya que se almacenan en un único sistema informático y se gestionan. Es más fácil comunicar y coordinar para obtener datos más fiables y significativos.

- **Redundancia de los datos:**

Los datos están centralizados y se almacenan en una sola ubicación. No hay duplicación de datos ni irrelevancia de estos.

- **Seguridad de los datos**

Debido al almacenamiento de los datos en un sistema informático centralizado, la seguridad de los datos debe ser mayor. El sistema de gestión de bases de datos centralizado es más seguro y eficiente.

- **Escalabilidad y localización**

Se pueden añadir o eliminar nuevos sistemas informáticos en el sistema de gestión de bases de datos centralizado con mayor facilidad.

- **Portabilidad de los datos**

Los datos pueden transferirse fácilmente de un ordenador a otro porque se almacenan en un sistema de gestión de bases de datos centralizado

- **Menor coste y mantenimiento**

El sistema de base de datos centralizado es más barato en cuanto a instalación y mantenimiento que otros sistemas de gestión de bases de datos y requiere un único sistema de almacenamiento y todos los ordenadores conectados pueden acceder a los datos. **Desventajas del sistema de bases de datos centralizado**

- **Procesamiento lento**

En el sistema de gestión de bases de datos centralizadas, los datos se almacenan en una sola ubicación y su velocidad de acceso y procesamiento es menor que la de otros sistemas de gestión. Se requiere más tiempo para acceder a los datos desde una sola ubicación.

- **Menor eficiencia**

En el sistema de gestión de bases de datos centralizadas, los datos se almacenan en una sola ubicación y su velocidad de acceso y procesamiento es menor que la de otros sistemas de gestión. Se requiere más tiempo para acceder a los datos desde una sola ubicación.

- **Pérdida de datos**

En un sistema de gestión de bases de datos centralizado, si se produce algún fallo en el sistema o se pierde algún dato, éste no se recupera.

5. Modelos arquitectónicos para DBMS distribuidos

- 5.1 Arquitectura cliente-servidor para DDBMS. Se trata de una arquitectura de dos niveles en la que la funcionalidad se divide en servidores y clientes. Las funciones del servidor abarcan principalmente la gestión de datos, el procesamiento de consultas, la optimización y la gestión de transacciones. Las funciones del cliente incluyen principalmente la interfaz de usuario. Sin embargo, tienen algunas funciones como la comprobación de la consistencia y la gestión de las transacciones. Las dos arquitecturas cliente-servidor son diferentes.
- 5.2 Arquitectura Peer- to-Peer para DDBMS. En estos sistemas, cada par actúa tanto como cliente como servidor para impartir servicios de base de datos. Los pares comparten sus recursos con otros pares y coordinan sus actividades. Esta arquitectura suele tener cuatro niveles de esquemas.
- 5.3 Arquitecturas Multi - DBMS.
 - Nivel de vista de la base de datos múltiple: Representa múltiples vistas de usuario que comprenden subconjuntos de la base de datos distribuida integrada.
 - Nivel conceptual de la base de datos múltiple: Representa la base de datos múltiple integrada que comprende las definiciones de la estructura lógica global de la base de datos múltiple.
 - Nivel interno de la base de datos múltiple: Describe la distribución de los datos en los diferentes sitios y la asignación de la base de datos múltiple a los datos locales.
 - Nivel de vista de la base de datos local: Describe la vista pública de los datos locales.
 - Nivel conceptual de la base de datos local: Describe la organización de los datos locales en cada sitio.
 - Nivel interno de la base de datos local: Describe la organización física de los datos en cada sitio.

6. Autonomía

La antinomia se refiere a la distribución del control no de los datos. Cada nodo tiene un cierto grado de control, Cada administrador local puede tener un nivel de autonomía distinto.

- Las operaciones locales no deben verse afectadas por la participación del nodo en un sistema distribuido.
- La forma de consulta global multibase no debe afectar individual
- El funcionamiento del sistema no debe verse afectado porque el sistema (nodos que lo conforman) se modifique

Desde la perspectiva de los usuarios, los datos se integran lógicamente en una base de datos. En estos sistemas estrechamente integrados, los gestores de datos se implementan de modo que uno de ellos tenga el control del procesamiento de cada solicitud de usuario, incluso si esa solicitud es atendida por más de un administrador de datos. Los administradores de datos no suelen funcionar como DBMS independientes a pesar de que suelen tener la funcionalidad para hacerlo.

- Anatomía de diseño, pueden utilizar los modelos de datos que decidan y técnicas de gestión
- Autonomía de comunicación, deciden que información comparten con el resto de nodos
- Autonomía de ejecución, pueden ejecutar las transacciones de la forma que desee.

7. Distribución

Por supuesto, estamos considerando la distribución física de datos en varios sitios; como comentamos anteriormente, el usuario ve los datos como un grupo lógico. Hay varias formas de distribuir los DBMS. Resumimos estas alternativas en dos clases: distribución cliente/servidor y distribución peer-to-peer

En la distribución cliente/servidor las tareas de comunicación se comparten entre las máquinas cliente y los servidores. Los DBMS cliente/servidor representan un compromiso práctico para distribuir la funcionalidad.

En los sistemas peer-to-peer, cada máquina tiene una funcionalidad completa de DBMS y puede comunicarse con otras para ejecutar consultas y transacciones. La mayor parte de los primeros trabajos sobre sistemas de bases de datos distribuidas han asumido una arquitectura peer-to-peer.

8. Heterogeneidad

Por tanto, cuando los manejadores de las bases de datos componentes son diferentes se trata de un Sistema multibase de datos heterogéneo. Existen dos casos de heterogeneidad a nivel Manejador de base de datos: a) El Sistema Multibase de datos está compuesto por diferentes Manejadores de base de Datos, pero pertenecen al mismo modelo de datos. Por ejemplo, Sybase y Oracle ofrecen Manejadores de Bases de Datos Relacionales. b) El Sistema Multibases de datos está compuesto por diferentes Manejadores de Base de Datos pertenecientes a diferentes modelos de datos, como el relacional y el objeto-relacional con Sybase Adaptive Server Enterprise y Versant Object Database por ejemplo. Tipos de heterogeneidad Los tipos de heterogeneidad a nivel sistemas de base de datos pueden dividirse en aquellos derivados de las diferencias en los Manejadores de Base de Datos y aquellos derivados de las diferencias en la semántica de los datos. Heterogeneidades por Diferentes Manejadores de Datos a) Componentes del Modelo de datos: Estructura, restricciones, lenguajes de consulta b) Implementación propia del manejador: manejo de transacciones, control de concurrencia, protocolos de respaldo y recuperación, etc. c) Diseño lógico de base de datos Heterogeneidad semántica d) Homónimos, sinónimos, conflictos de dominio, manejo de información faltante, conflictos de

9. Alternativas arquitectónicas

Arquitectónicamente, un sistema de base de datos distribuida consiste en un conjunto (posiblemente vacío) de sitios de consulta y un conjunto no vacío de sitios de datos. Los sitios de datos tienen capacidad de almacenamiento de datos, mientras que los sitios de consulta no. Estos últimos solo ejecutan la interfaz de usuario (además de las aplicaciones) para facilitar el acceso a los datos en los sitios de datos.

En el mundo real, se pueden utilizar varias alternativas arquitectónicas diferentes en lugar de la DDBMS tradicional y homogéneo, que está diseñado de arriba hacia abajo. Bases de datos federadas o las bases de datos múltiples son servidores de bases de datos existentes integrados que se construyen desde abajo arriba. En un sistema de bases de datos federado los datos que necesita una organización ya existe en múltiples servidores de bases de datos. Estas bases de datos pueden o no ser del mismo tipo (mismo modelo de datos y/o lenguaje de modelado de datos).

10. Sistemas cliente/servidor

La idea general de es muy simple y elegante : distingue la funcionalidad que necesita ser proporcionada y divide estas funciones en 2 clases: Funciones del cliente y funciones del servidor. Esto provee una arquitectura de dos niveles lo cual hace más fácil de manejar la complejidad de los DBMSs y la complejidad de distribución.

Nosotros nos enfocamos en que software debe de correr y que software debe correr en el servidor de la máquina, a esto nos referimos a un sistema cliente servidor en bases de datos. En esta arquitectura nos encontramos habitualmente con una serie de En este modelo de informática distribuida, existe una comunicación entre la aplicación cliente, que requiere la información, y el servidor de base de datos, que se la proporciona. De esa forma, la aplicación cliente se especializa en lo que conocemos como interfaz de usuario y el servidor se dedica a las tareas de gestión de la base de datos.

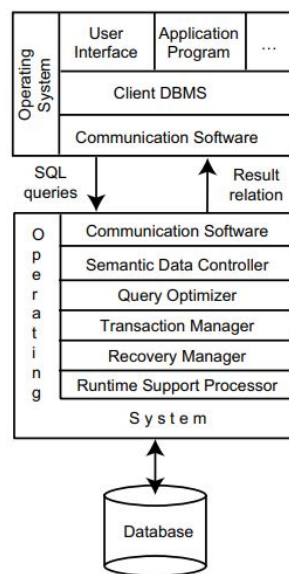


Figura 2: Arquitectura Cliente/Servidor

11. Sistema peer to peer

Peer-to-peer o P2P, también conocida en castellano como “red entre pares” o “de igual a igual”, se refiere a las redes de ordenadores que usan una arquitectura distribuida. Eso significa que todos los ordenadores o dispositivos que forman parte de ella comparten las cargas de trabajo en la red. Los ordenadores o dispositivos electrónicos que forman parte de una red peer-to-peer se llaman pares o peers. Cada par es igual a los demás, y cada par tiene los mismos derechos y obligaciones que los demás. Los pares son clientes y servidores al mismo tiempo.

Además, cada recurso disponible en una red punto a punto se comparte entre los pares, sin que intervenga ningún servidor central. Los recursos compartidos en una red P2P pueden ser cosas como el uso del procesador, el espacio de almacenamiento en disco o el ancho de banda de la red.

- **Alto grado de descentralización:**

Los pares implementan la funcionalidad del cliente y del servidor y la mayoría del estado del sistema y las tareas se asignan dinámicamente entre los pares. Hay pocos o ningún nodo dedicado con estado centralizado. Como resultado, la mayor parte del cómputo, el ancho de banda

y el almacenamiento necesarios para operar el sistema son aportados por los nodos participantes.

- **AUTO ORGANIZACIÓN:**

Una vez que se introduce un nodo en el sistema (normalmente proporcionándole la dirección IP de un nodo participante y cualquier material clave necesario), se necesita poca o ninguna configuración manual para mantener el sistema.

- **MÚLTIPLES DOMINIOS ADMINISTRATIVOS**

Los nodos participantes no son propiedad ni están controlados por una sola organización. En general, cada nodo es propiedad y está operado por un individuo independiente que voluntariamente se une al sistema

12. Arquitectura de un sistema de base de datos múltiple

Los sistemas de bases de datos múltiples (MDBS) representan el caso en el que los DBMS individuales (ya sea distribuidos o no) son totalmente autónomos y no tienen concepto de cooperación; que puede ni siquiera “saben” de la existencia del otro o cómo hablar entre ellos.

Referencias

- [1] Boylestad, Robert y Louis Nashelsky: *Electrónica: Teoría de circuitos y dispositivos electrónicos*. Pearson Hall, 10ª edición, 2009.
- [2] Floyd, Thomas y R.N. Salas: *Dispositivos electrónicos 8ED*. Pearson Educación, 2008.
- [3] EcuRed: *Filtro electrónico - EcuRed*. https://www.ecured.cu/Filtro_electr%C3%B3nico.
- [4] *Ventajas y desventajas de los filtros activos*. <http://gigatecno.blogspot.com/2013/02/ventajas-y-desventajas-de-los-filtros.html>.
- [5] Torres, Hector: *Filtro pasa bajas pasivo en 5 minutos*, Diciembre 2017. <https://hetpro-store.com/TUTORIALES/pasa-bajas-filtro/>.
- [6] undefined: *Filtro Butterworth*, Octubre 2017. <https://www.youtube.com/watch?v=NEs5KBsU9b4>.
- [7] *FILTROS ACTIVOS*. <https://www.angelfire.com/linux/fiet601/segundo/FA.htm>.