



TIPOS DE FRAGMENTACIÓN EN BASES DE DATOS DISTRIBUIDAS

Resumen / Abstract

En el trabajo se reflejan aspectos importantes referentes al diseño de una base de datos distribuida. Se profundiza en el problema de la fragmentación de los datos y se brindan algunos de los algoritmos para fragmentación horizontal, fragmentación horizontal derivada y fragmentación vertical. El tema seleccionado es parte de una investigación de tesis de maestría que presenta el desarrollo de un modelo arquitectónico para la Plataforma de Gestión Comercial en la Unidad de Negocios Internacionales de ETECSA.

In this paper appear important things that approach the distributed database design. This work deepens in the problem of the fragmentation data and offers some algorithms for horizontal fragmentation, derived horizontal fragmentation and vertical fragmentation. The selected topic is part of an investigation of a master's thesis that presents the development of an architectonic model for the Commercial Management Platform in the ETECSA International Trade Center.

Palabras clave / Key words

Base de datos, base de datos distribuida sistema de base de datos distribuida fragmentación de datos, arquitectura de las bases de datos distribuidas

Databases, distributed database, distributed database systems, fragmentation data, distributed database architecture

INTRODUCCIÓN

Una base de datos distribuida (BDD) es una colección de datos que pertenecen lógicamente al mismo sistema, pero que están distribuidos sobre diferentes ordenadores de la red. Esta definición enfatiza dos aspectos importantes en una BDD:¹

- **Distribución:** Los datos no residen en el mismo lugar. De este modo se puede distinguir una BDD de una base de datos (BD) centralizada.

- **Correlación lógica:** Es decir, el hecho por el cual los datos tienen algunas propiedades o características que los relaciona. De este modo se puede distinguir una BDD de un conjunto de BD locales o ficheros residentes en diferentes lugares de una red de ordenadores.

En la arquitectura de una BDD se pueden identificar tres capas: **vista de usuario, conceptual** y **física**. En la capa de usuario se encuentran todas las aplicaciones, las pantallas de entrada de cada dato y los reportes requeridos para la empresa para la cual se representa el modelo. En la capa conceptual está el modelo del negocio subyacente, usualmente el modelo entidad-relación. En la capa física está el modelo físico y la estructura de la base de datos. Desde la perspectiva de una BDD, los usuarios desde la capa de usuario examinan y manipulan los datos como si la base de datos estuviese centralizada en ese nodo. Desde el punto de vista físico la BD está fragmentada y ubicada en diferentes sitios.²

Entre los beneficios que presentan las bases de datos distribuidas, se encuentran:

- Los datos son localizados en un 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 y existe una autonomía e independencia entre los nodos.
- Los datos se pueden colocar físicamente en el lugar donde se acceden con mayor frecuencia, haciendo que los usuarios tengan control local de los datos con los que interactúan. Esto resulta en una autonomía local.
- Mediante la replicación de información, las bases de datos distribuidas pueden presentar cierto grado de tolerancia a fallas haciendo que el funcionamiento del sistema no dependa de un solo lugar.

ARQUITECTURA DE REFERENCIA PARA LAS BD DISTRIBUIDAS

Los elementos que forman la arquitectura son los siguientes:

Esquema global: Define todos los datos que están contenidos en la BDD como si la BD no fuese distribuida. Por esta razón, el esquema global puede ser definido exactamente de la misma manera que una BD no distribuida. Refiriéndose a un modelo relacional, se tendrían relaciones globales.

Fragmentos: Cada relación global puede ser dividida en porciones que no se solapen, llamados fragmentos. El mapa resultante se denomina esquema de fragmentación. Una relación global puede dividirse en n fragmentos y un fragmento solo puede pertenecer a una relación global. Los fragmentos se referencian por un nombre de relación global y un subíndice. Por ejemplo R_i indica el fragmento i de la relación R .

Los fragmentos son porciones lógicas de relaciones globales que pueden estar físicamente ubicadas en uno o varios nodos de la red. El esquema de ubicación define en qué nodos va a ser almacenado un fragmento. El tipo de mapa definido en el esquema de ubicación determina si la BDD es redundante o no. Todos los fragmentos que corresponden a la misma relación global R y están ubicados en el mismo nodo constituyen la imagen física de una relación global R en el nodo. Las imágenes físicas se pueden referenciar por el nombre de la relación global R y un superíndice; por ejemplo, R^j indica la imagen física de la relación global R en el nodo j .

Por último, se denomina copia de fragmento a la información de un fragmento en un nodo determinado, y se denota usando el nombre de la relación global, un subíndice y un superíndice. Ejemplo: R^3_2 indica copia del fragmento R_2 que está ubicado en el nodo 3. Dos imágenes físicas pueden ser idénticas; se dice entonces que una imagen física es copia de otra imagen física.

En un nivel más bajo, es necesario construir un mapa que relacione las imágenes físicas con los objetos que son manipulados por los gestores locales. Este mapa se llama esquema de mapas locales y depende del tipo de SGBD local. Por lo tanto, en un sistema heterogéneo se tienen diferentes tipos de mapas locales en los distintos nodos.

FRAGMENTACIÓN

Los principales problemas de la fragmentación² se pueden resumir básicamente en:

- Encontrar la unidad apropiada de distribución, es decir, definir qué contiene un fragmento.
- El rendimiento se afecta cuando existen aplicaciones que necesitan tener una vista completa de un objeto o entidad (relación, en el modelo relacional) y está descompuesta en fragmentos ubicados físicamente en sitios distintos. Esta recuperación requiere la ejecución de operaciones de unión y combinación.
- Se pierde el significado semántico del objeto o entidad al tenerse el concepto subdividido en fragmentos ubicados en diferentes sitios. Esto puede provocar, por ejemplo, complicaciones en la interpretación y verificación del modelo conceptual que representa a los requerimientos.

A pesar de estos inconvenientes, la fragmentación facilita el proceso concurrente de las transacciones y, por lo tanto, la recuperación de información. Con el fin de realizar una fragmentación adecuada es necesario proporcionar información que ayude a realizarla. Esta información normalmente debe ser proporcionada por el usuario y tiene que ver con cuatro tipos: 1. Información sobre el significado de los datos; 2. Información sobre las aplicaciones que los usan; 3. Información acerca de la red de comunicaciones; 4. Información acerca de los sistemas de cómputo.

Existen tres tipos de fragmentación: 1. Fragmentación horizontal; 2. Fragmentación vertical; 3. Fragmentación mixta o híbrida. Dado que una relación se corresponde esencialmente con una tabla y la cuestión consiste en dividirla en fragmentos menores, inmediatamente surgen dos alternativas lógicas para llevar a cabo el proceso: la división horizontal y la división vertical. Estos dos tipos de partición podrían considerarse los fundamentales y básicos. Sin embargo, existen otras alternativas. Fundamentalmente, se habla de fragmentación mixta o híbrida cuando el proceso de partición hace uso de los dos tipos anteriores. A continuación se enuncian las tres reglas que se han de cumplir durante el proceso de fragmentación, las cuales asegurarán la ausencia de cambios semánticos en la base de datos durante el proceso.^{3,4}

Condición de completitud: La descomposición de una relación R en los fragmentos R_1, R_2, \dots, R_n es completa si y solamente si cada elemento de datos en R se encuentra en uno o varios fragmentos R_i . Esta propiedad asegura que los datos de la relación global se proyecten sobre los fragmentos sin pérdida alguna.

Condición de reconstrucción: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , entonces debe existir algún operador relacional ∇ , tal que, $R = \nabla_{1 \leq i \leq n} R_i$. La reconstrucción de la relación a partir de sus fragmentos asegura la preservación de las restricciones definidas sobre los datos en forma de dependencias.

Condición de fragmentos disjuntos: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , y un elemento de datos d_i está en algún fragmento R_j , entonces, no debe estar en ningún otro fragmento R_k ($k \neq j$). Esta regla asegura que los

fragmentos horizontales sean disjuntos. Si una relación R se descompone verticalmente, sus atributos primarios clave normalmente se repiten en todos sus fragmentos.

Fragmentación horizontal

La fragmentación horizontal se realiza sobre las tuplas de la relación. Cada fragmento será un subconjunto de las tuplas de la relación. Existen dos variantes de la fragmentación horizontal: la primaria y la derivada. La fragmentación horizontal primaria de una relación se desarrolla empleando los predicados definidos en esa relación. Por el contrario, la fragmentación horizontal derivada consiste en dividir una relación partiendo de los predicados definidos sobre alguna otra.

Para poder construir una fragmentación, es necesario proporcionar información acerca de la base de datos y acerca de las aplicaciones que las utilizan. En primer término, es necesario proporcionar la información acerca del esquema conceptual global. En este sentido es importante dar información acerca de las relaciones que componen la base de datos, la cardinalidad de cada relación y las dependencias entre relaciones.⁴

En segundo lugar, se debe proporcionar información acerca de la aplicación que utiliza la base de datos. Esta puede ser tanto cualitativa como cuantitativa. La primera guiará la fragmentación, mientras que la cuantitativa se necesitará en los modelos de asignación. La principal información de carácter cualitativo son los predicados empleados en las consultas de usuario. Dada una relación $R(A_1, A_2, \dots, A_n)$, donde A_i es un atributo definido sobre el dominio D_i , un predicado simple p_j definido en R tiene la forma $p_j: A_i \theta$. Valor donde $\theta \in \{=, <, \neq, \leq, >, \geq\}$ y Valor $\in D_i$. Para la relación R se define un conjunto de predicados simples como $P_r = \{p_1, p_2, \dots, p_m\}$

Sobre la información cuantitativa necesaria relativa a las aplicaciones, se definen dos conjuntos de datos.⁴

Selectividad minitérmino: Es el número de tuplas de una relación a las que accede una consulta de acuerdo con un predicado minitérmino dado. Se denota la selectividad de un minitérmino m_i como $sel(m_i)$.

Frecuencia de acceso: Es la frecuencia con la que un usuario accede a los datos. Si $Q = \{q_1, q_2, \dots, q_q\}$ es un conjunto de consultas de usuario, $acc(q_i)$ indica la frecuencia de acceso a la consulta q_i en un periodo dado.

Fragmentación vertical

Una fragmentación vertical³ de una relación R produce fragmentos R_1, R_2, \dots, R_r , cada uno de los cuales contiene un subconjunto de los atributos de R así como la llave primaria de R . El objetivo de la fragmentación vertical es particionar una relación en un conjunto de relaciones más pequeñas de manera que varias de las aplicaciones de usuario se ejecutarán sobre un fragmento. En este contexto, una fragmentación **óptima** es aquella que produce un esquema de fragmentación que minimiza el tiempo de ejecución de las consultas de usuario.

La fragmentación vertical ha sido estudiada principalmente dentro del contexto de los sistemas de manejo de bases de datos centralizados como una herramienta de diseño, la cual permite que las consultas de usuario traten con relaciones más pequeñas

haciendo, por tanto, un número menor de accesos a páginas. Es más complicada que el particionamiento horizontal, ya que existe un gran número de alternativas para realizarla. Por lo tanto, se utilizan heurísticas para hacer el particionamiento. Los dos enfoques básicos son:

1. Agrupamiento: Inicia asignando cada atributo a un fragmento, y en cada paso, algunos de los fragmentos satisfaciendo algún criterio se unen para formar un solo fragmento.

2. División: Inicia con una sola relación realizar un particionamiento basado en el comportamiento de acceso de las consultas sobre los atributos.

Dado un conjunto de consultas $Q = \{q_1, q_2, \dots, q_q\}$ que serán aplicadas a la relación $R[A_1, A_2, \dots, A_n]$, se define la función

$$use(q_i, A_j) = \begin{cases} 1 & \text{Si el atributo } A_j \text{ es referido por la consulta } q_i \\ 0 & \text{en caso contrario} \end{cases}$$

Los vectores $use(q_i, \cdot)$ son fáciles de definir si el diseñador conoce las aplicaciones que serán ejecutadas en la base de datos. La medida de afinidad entre dos atributos A_i y A_j de una relación $R[A_1, A_2, \dots, A_n]$ con respecto al conjunto de consultas $Q = \{q_1, q_2, \dots, q_q\}$ se define como sigue: $aff(A_i, A_j) = \Sigma$ las consultas que acceden A_i y A_j Si $(refl(q_k) acc(q_k))$, donde, $refl(q_k)$ es el número de accesos a los atributos (A_i, A_j) para cada ejecución de la consulta q_k en el sitio S_i y $acc_i(q_k)$ es la frecuencia de acceso de la consulta previamente definida y modificada para incluir las frecuencias en sitios diferentes.

Fragmentación mixta o híbrida

La forma más simple de construir fragmentación mixta consiste en aplicar fragmentación horizontal a fragmentos verticales, o aplicar fragmentación vertical a fragmentos horizontales. Por ejemplo:

a) Fragmentación vertical seguida por una fragmentación horizontal

A1	A2	A3	A4	A5

b) Fragmentación horizontal seguida por una fragmentación vertical

A1	A2	A3	A4	A5

Aunque se pueden generar árboles de fragmentación muy complejos, normalmente debe ser suficiente con dos niveles. La segunda fragmentación puede ser aplicada a subconjuntos de fragmentos producidos por la primera. En el ejemplo anterior: a) La fragmentación horizontal se aplica a un solo fragmento producido por la fragmentación vertical; b) La fragmentación vertical se aplica a un solo fragmento producido por la fragmentación horizontal.

ALTERNATIVAS SOBRE REPLICACIÓN PARA LA ASIGNACIÓN DE FRAGMENTOS

Suponiendo que los datos son fragmentados correctamente, habrá que decidir sobre la manera de asignar los fragmentos a los distintos sitios de la red. Cuando una serie de datos se asignan, estos pueden replicarse para mantener una copia. Las razones para la réplica giran en torno a la seguridad y a la eficiencia de las consultas de lectura. Si existen muchas reproducciones de un elemento de datos, en caso de fallo en el sistema se podría acceder a esos datos ubicados en sitios distintos. Además, las consultas que acceden a los mismos datos pueden ejecutarse en paralelo, ya que habrá copias en diferentes sitios. Por otra parte, la ejecución de consultas de actualización, de escritura, implicaría la actualización de todas las copias que existan en la red, cuyo proceso puede resultar problemático y complicado.

Por tanto, un buen parámetro para afrontar el grado de réplica sería la cantidad de consultas de lectura que se efectuarán, así como el número de consultas de escritura que se llevarán a cabo. En una red donde las consultas que se procesen sean mayoritariamente de lectura, se podrá alcanzar un alto grado de réplica, no así en el caso contrario.

Una base de datos fragmentada es aquella donde no existe réplica alguna. Los fragmentos se alojan en sitios donde únicamente existe una copia de cada uno de ellos a lo largo de toda la red. En caso de réplica, se puede considerar una base de datos totalmente replicada, donde existe una copia de todo el banco de datos en cada sitio, o considerar una base de datos parcialmente replicada donde existan copias de los fragmentos ubicados en diferentes sitios. El número de copias de un fragmento será una de las posibles entradas a los algoritmos de asignación, o una variable de decisión cuyo valor lo determine el algoritmo.⁴

CONCLUSIONES

Con el desarrollo de este trabajo se ha logrado realizar una búsqueda bibliográfica acerca de las bases de datos distribuidas, fundamentalmente en el tema de la fragmentación de los datos y su ubicación en los nodos que componen la red. Esta revisión ha permitido a la autora centrarse en los elementos fundamentales para lograr el diseño más óptimo de los datos a distribuir y(o) replicar en cada punto del sistema distribuido. 

REFERENCIAS

1. IGLESIAS, EVA L.: *Bases de datos distribuidas*, ESEI, Universidad de Vigo. <http://trevinca.ei.uvigo.es/~evali/abd/BDDWeb.pdf>, (abril, 2005)
2. HERNÁNDEZ GONZÁLEZ, ANAISA. "Base de datos distribuida", Maestría en Informática Aplicada, CEIS, Ciudad de La Habana, marzo, 2005.
3. Cinvestav. *Diseño de Bases de datos distribuidas*. http://www.cs.cinvestav.mx/SC/prof_personal/adiaz/Disdb/Cap_3.html, abril, 2005.
4. RODRÍGUEZ SANTOS, JORGE: "Diseño de bases de datos distribuidas", <http://usuarios.lycos.es/jrodr35/index.htm>, abril, 2005.



Ediciones e Imprenta

Con más de 20 años de experiencia en la actividad editorial, el Departamento de Ediciones e Imprenta cuenta con un personal altamente calificado que le garantizará el desarrollo de su trabajo.

SERVICIOS QUE OFERTAMOS

Edición, diseño e impresión

Revistas, Libros, folletos, plegables, afiches, agendas, tarjetas de presentación, credenciales, blocks de notas, modelos, invitaciones

Encuadernación

Encuadernado de folletos, trabajos de diplomas, tesis doctorales, etcétera.

Solicitud de ISBN

Entre otros servicios, publicamos las revistas científicas en formato digital en los sitios Web de la Cujae (intranet o Internet). Usted puede acceder a estos sitios a través de las siguientes direcciones:
<http://intranet.cujae.edu.cu/ediciones/>
www.cujae.edu.cu/ediciones

CONTÁCTENOS EN

Instituto Superior Politécnico José Antonio Echeverría,
 Calle 114, No. 11901, e/ 119 y 127, Marianao,
 Ciudad de La Habana.

 266 3699, 266 3701  mail: hamigo@tesla.cujae.edu.cu