

Sistemas de Bases de Datos Federadas

Luis Fernando Espino Barrios
Instituto Tecnológico de Costa Rica
luisespino@yahoo.com
Octubre 2009

Resumen: Este artículo pretende introducir conceptualmente al lector en los sistemas de bases de datos federadas, que son colecciones de componentes que cooperan entre sí manteniendo su propia autonomía. También se muestran las características que debe tener un sistema federado, de las cuales las principales son la heterogeneidad y la autonomía; se presentan también varios enfoques de arquitectura basados en esquemas de diferentes niveles, así como el funcionamiento que se concreta en tareas específicas de desarrollo y de operación, por último se presentan algunos enfoques de implementación haciendo énfasis en sus diferencias.

Palabras clave: Bases de datos federadas, bases de datos distribuidas, sistemas administrativos de bases de datos federadas, clúster.

1. Introducción

Una definición simple y de las más antiguas [1], dice que los sistemas de bases de datos federadas son colecciones de componentes cooperativos pero autónomos de sistemas de bases de datos convencionales.

Una definición más reciente [2] establece que un sistema de bases de datos federadas es un sistema múltiple de base de datos, en el cual, cada nodo en la federación mantiene su autonomía en los datos y define un conjunto de esquemas de exportación, a través de los cuales se hacen disponibles los datos a otros nodos.

El termino federación [3] se refiere a la colección de bases de datos constituyentes que participan en una base de datos federada.

De forma análoga, un sistema administrativo de bases de datos (DBMS) que es la herramienta de software que provee la administración de las bases de datos centralizadas, también existe su equivalente en ambientes federados, llamado sistema administrativo de bases de datos federado (FDBMS), con ciertas diferencias y que tiene puntos desafiantes, tales como las ejecuciones de transacciones y la concurrencia, manteniendo la consistencia de la base de datos.

En el resto del artículo se detallan las características que un sistema federado debe tener, la arquitectura, el modelo de datos, el funcionamiento, algunas diferencias entre sistemas de bases de datos y ciertas implementaciones para sistemas de bases de datos federadas.

2. Características

Dependiendo del enfoque, un sistema federado debe cumplir con ciertas características, por ejemplo, en [1] se presentan las siguientes:

- Distribución: Los datos pueden estar ubicados entre múltiples bases de datos.
- Heterogeneidad: Se debe permitir diferencias en el hardware, software y en los sistemas de comunicación.
- Heterogeneidad de semántica: Ocurre cuando hay discrepancias acerca del significado, interpretación o pretensión de utilización de los mismos datos o datos relacionados.
- Autonomía: Se define como la capacidad de manejar su propio sistema de base de datos, es decir, que tengan control separado e independiente.

Aunque hay que destacar que no solo esas características puede tener un sistema federado, a continuación se nombran otros enfoques, así como ciertos desafíos que alcanzar. Por ejemplo, en [4] se menciona que las características de heterogeneidad y autonomía en sistemas de bases de datos federadas causa una variedad de dificultades en el procesamiento de consultas globales y en la correcta ejecución de las transacciones que deben satisfacer la seriabilidad global.

Otra característica adicional que se menciona en [5] es la inter operatividad, que es un condición mediante la cual sistemas heterogéneos pueden interactuar entre sí. La inter operatividad entre componentes de sistemas de bases de datos es alcanzar por medio de la capacidad de componentes individuales el compartir e intercambiar unidades de información de manera activa y cooperativamente con otros componentes de la federación.

3. Arquitectura

Para bases de datos centralizadas la arquitectura ANSI/SPARC [6] se aplica bien, está compuesta de tres niveles: el esquema conceptual, que describe las estructuras de datos conceptuales o lógicos; el esquema interno, describe las características físicas de la estructura de datos lógica del esquema conceptual; y el esquema externo, donde se accede por un usuario o una clase de usuarios.

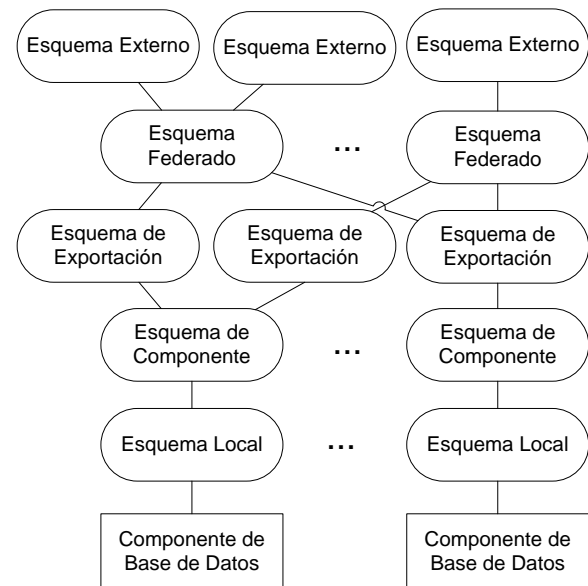


Figura 1: Arquitectura de cinco niveles propuesta por Sheth.

Sin embargo, esta arquitectura no es adecuada para sistemas federados, por lo que se debe extender, hay autores que extienden la arquitectura a cuatro niveles, otros que solamente cambian la estructura y en [1] se extiende a cinco niveles, como se muestra en la (Figura 1: Arquitectura de cinco niveles propuesta por Sheth. Los cinco esquemas definidos son:

- Esquema local: Es un esquema conceptual de un componente de base de datos.
- Esquema de componente: Es derivado traduciendo el esquema local en un modelo de datos común (CDM) o canónico.
- Esquema de exportación: Representa un subconjunto de un esquema de componente disponible en los sistemas de bases de datos federadas.
- Esquema federado: Es una integración de múltiples esquemas de exportación.
- Esquema externo: Define un esquema para un usuario o aplicación o clase de usuarios/aplicaciones

En [3] se propuso otra arquitectura, basada en tres niveles:

- Esquema privado: Describe la porción de los componentes de datos que es local al componente.
- Esquema de exportación: La porción de esquemas de exportación de un componente que especifica la información que el componente esté dispuesto a compartir con otros componentes de la federación.
- Esquema de importación: El esquema de importación de un componente especifica la información que el componente desea utilizar de otro componente.

En [3] se mencionó que hay dos requerimientos conflictivos que hay que resolver, debido a la ausencia de una autoridad central: uno es que los componentes deben mantener su autonomía como sea posible, y otro es que los componentes deben estar habilitados para alcanzar cierto grado de intercambio de información.

4. Modelo de datos

En la sección de arquitectura se mencionó el modelo de datos común (CDM) o canónico, hay dos razones para definir esquemas de componentes en CDM: la primera es porque describen los esquemas locales divergentes usando una representación sencilla, y la segunda es que la semántica que se ha perdido en los esquemas locales, se puede adicionar en los esquemas de componentes.

En [5] se propuso el Modelo de Datos Semántico Heterogéneo (HSDM), es un modelo de datos orientado a objetos que tiene una semántica enriquecida y expresiva, facilitando el intercambio de información y componentes federados produciendo esquemas conceptuales. Por lo que las bases de datos HSDM son colecciones de objetos y relaciones entre ellos.

Otro enfoque [7] es el modelo de administración de metadatos para bases de datos federadas utilizando una base de datos relacional como un repositorio central de metadatos. Se utilizó un modelo orientado a objetos para representar las bases de datos, las tablas, las columnas, los documentos, los usuarios y los programas como objetos enlazados por asociaciones. Se utiliza SQL para armar objetos y crear asociaciones en parejas de objetos basados en tipos de datos definidos.

En [3] se propuso un modelo de datos orientado a objetos, llamado modelo evento, este modelo evento es característicamente semántico, debido a su definición y está basado en las tres primitivas que se presentan a continuación:

- Objetos: Es un elemento básico de modelado que corresponde al mundo real como una entidad o un concepto, dividiéndose en abstractos y descriptores.

- Tipos: Son colecciones de objetos variantes en el tiempo que comparten propiedades, los objetos de un tipo dado se llaman instancias.
- Mapas: Son funciones que mapean objetos de un tipo de dominio a conjuntos de objetos en el conjunto potencia de cierto tipo de rango

5. Funcionamiento

Hay dos tipos de tareas que describen el funcionamiento de los sistemas de bases de bases de datos federadas, las tareas de desarrollo y las de operación.

5.1. Tareas de desarrollo

En [1] se definen cuatro tareas de desarrollo:

- Traducción de esquemas: Se ejecuta cuando un esquema representado en un modelo de datos es mapeado a un esquema equivalente representado en diferente modelo de datos.
- Control de acceso: Un sistema federado debe estar diseñado para controlar el acceso a los componentes de la base de datos por usuarios federados.
- Negociación: Es el dialogo entre dos administradores para alcanzar un acuerdo respecto a los esquemas de exportación y a las operaciones permitidas, se debe hacer por medio de un protocolo para el intercambio de mensajes.
- Integración de esquemas: Se refiere a la integración de múltiples vistas de usuarios en un solo esquema, es decir, integrar esquemas en un solo esquema federado integrando esquemas de exportación por medio de *bottom-up*.

5.2. Tareas de operación

De la misma manera, en [1] se mencionan otras cuatro tareas que corresponden a la operación:

- Formulación de consultas: El lenguaje de consultas puede ser el mismo que se utiliza para bases de datos centralizados, debido a que las bases de datos federadas son transparentes en ese aspecto.
- Transformación de comandos: Debe existir un procesador de transformaciones de comandos, que traduce esos comandos en un lenguaje, llamado lenguaje origen, a otro lenguaje, llamado lenguaje destino.
- Procesamiento de consultas y optimización: El procesamiento implica convertir una consulta de un esquema federado a un esquema de exportación y luego ejecutarlas. Respecto a los procesos de optimización y de procesamiento son similares a los de las bases de datos distribuidas.
- Administración de transacciones globales: Es el responsable de mantener la consistencia entre las bases de datos, mientras se permita cierta concurrencia a través de múltiples bases de datos, aunque esta tarea es muy complicada en ambientes heterogéneos.

Con el correcto funcionamiento se garantiza la consistencia de la base de datos federada, aunque hay otro desafío, el cual es la concurrencia, en donde se debe garantizar la ejecución serial de las transacciones tanto locales como globales. Se han propuesto varias soluciones aunque con poco grado de concurrencia y con posibilidad de caer en interbloqueos.

En [8] se propuso un enfoque novedoso que provee alta concurrencia y reduce el gasto de recursos del sistema, mientras se mantiene la seriabilidad global. Este enfoque es capaz de ajustar dinámicamente el orden de la seriabilidad global para concordar con el orden local, aceptando más ejecuciones y evitando interbloqueos globales.

6. Diferencias

En [3] se presentó una clasificación de bases de datos, inicialmente se mencionan dos dimensiones:

- Estructura conceptual/lógica
- Estructura y organización física

Adicionalmente, cada dimensión se puede dividir en dos partes:

- Centralizada
- Descentralizada

Con base a las clasificaciones anteriores, se puede realizar la siguiente clasificación de bases de datos:

- Las bases de datos que son lógicas y físicamente centralizadas pertenecen a las bases de datos integradas convencionales.
- Las bases de datos que son lógicamente centralizadas y físicamente descentralizadas son llamadas bases de datos distribuidas o compuestas.
- Las bases de datos lógicamente descentralizadas y ya sea físicamente centralizadas o físicamente descentralizadas representan las bases de datos federadas.

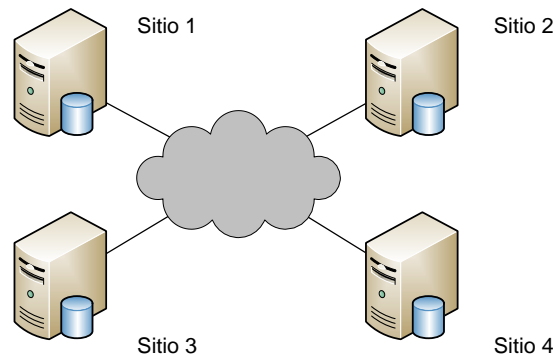


Figura 2: Ejemplo de un Sistema de Bases de Datos Distribuido.

Por lo que una base de datos federada es un caso especial de las bases de datos distribuidas, en la (Figura 2: Ejemplo de un Sistema de Bases de Datos Distribuido) se muestra una instancia de bases de datos distribuidas.

La diferencia entre estos dos sistemas de bases de datos radica que en las bases de datos federadas intervienen diferentes propietarios independientes que compartirán un esquema conceptual en común aunque tengan diferentes tipos de fuentes de datos, mientras que en las distribuidas se pretende realizar una fragmentación de los datos en esquemas similares.

Entre las principales diferencias están que la base de datos federada es un tipo de sistema centralizado que reúne sus datos de una federación de servidores heterogéneos, mientras que las distribuidas se pueden acceder desde cualquier servidor miembro.

En ambientes distribuidos se garantiza las transacciones, la concurrencia, la replicación, mientras que en las federadas no. Una instancia de bases de datos federadas se muestra en la (Figura 3: Ejemplo de un Sistema de Bases de Datos Federadas).

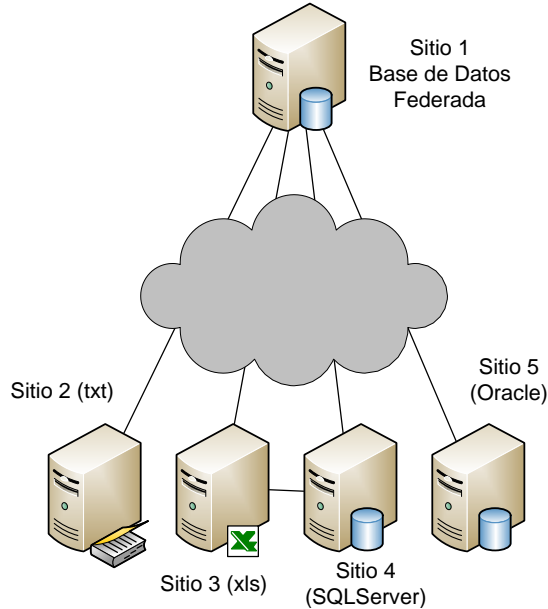


Figura 3: Ejemplo de un Sistema de Bases de Datos Federadas.

Y por último las bases de datos federadas es sencillo agregar un nodo más, debido a que el servidor maneja esa interacción, mientras que los sistemas distribuidos no es tan fácil agregar un nodo, debido a que hay que actualizar el direccionamiento de todos los nodos, es decir, cada nodo debe saber la ubicación del nuevo nodo.

7. Implementaciones

7.1. Remote-Exchange

Varios enfoques de *frameworks* se han propuesto, entre algunos de ellos sobresale el Remote-Exchange [9], es un proyecto de investigación que propone un enfoque y mecanismo para apoyar el intercambio del comportamiento entre los sistemas de bases de datos en una federación.

En el modelo se utilizan tres diferentes tipos de funciones: de almacenamiento, derivadas y computadas.

En la implementación se utilizaron componentes de bases de datos federadas de Omega e Iris. La importancia del enfoque radica en la separación de la ubicación de datos y de la ubicación de la ejecución de los métodos.

7.2. PEER

PEER [10], es un sistema federado de administración de información orientado a objetos, desarrollado para apoyar el intercambio de información a través de nodos cooperativos autónomos y heterogéneos.

Su arquitectura está basada por esquemas, cada nodo se representa por:

- Esquema local (LOC)
- Esquema de importación (IMP)
- Esquema de exportación (EXP)
- Esquema integrado (INT)

Su característica principal es la transparencia física y lógica de la distribución de información de los nodos a través del procesamiento de consultas federadas.

7.3. Myraid

Myraid [4] es un prototipo de sistema de bases de datos federadas desarrollado por la Universidad de Minnesota para satisfacer los orígenes de datos heterogéneos, las incompatibilidades a nivel de sistema y la falta de integración. Posee una arquitectura flexible que permite la administración de transacciones y procesamiento de consultas.

7.4. SQL Server

En SQL Server se puede implementar un servidor a través de particiones horizontales, se utiliza esta técnica para grandes bases de datos que consideran la federación como la manera de balancear el procesamiento a través de diferentes servidores, aunque su implementación requiere de nodos con SQL Server u servidores de bases de datos que implementen la partición horizontal.

En la biblioteca¹ en línea de SQL Server 2008 se plantea una capa de servidor federado que proporciona ciertas diferencias internas a comparación con los servidores centralizados:

- Hay una instancia ejecutándose de SQL Server en cada servidor miembro.
- Cada servidor miembro tiene una base de datos miembro, y los datos están propagados a través de las diferentes bases de datos.
- Las tablas de la base de datos original está particionada de manera horizontal en tablas miembro. Hay una tabla miembro por cada base de datos miembro, y las vistas particionadas y distribuidas son utilizadas para hacer que parezca como si hubiera una copia total de la tabla original en cada servidor miembro.
- La capa de aplicación debe estar habilitada para hallar sentencias SQL en el servidor miembro, que contengan la mayoría de datos referenciados por la sentencia.

¹ <http://msdn.microsoft.com/en-us/library/ms190381.aspx>

7.5. IBM Federated Database

Las capacidades de federación de IBM [11] están disponibles a través de una variedad de productos, entre los cuales se destaca DB2 UDB, DB2 DataJoiner, entre otros. Dichas herramientas proveen facilidades para combinar la información de múltiples fuentes de datos, implementando así las bases de datos federadas.

Entre algunas características importantes destacan la transparencia, la heterogeneidad, el alto grado de función, la extensibilidad, la autonomía y el rendimiento optimizado.

7.6. MySQL: Federated Store Engine

El motor de almacenamiento federado está disponible desde la versión 5.0.3, y sirve para acceder datos en tablas tanto de bases de datos remotos como locales.

Según la documentación de MySQL² se puede crear tablas federadas y tendrán la extensión .frm para tablas remotas y para tablas locales tendrán la extensión .myd. Para la lectura de datos se utiliza un API de cliente de MySQL, y utiliza un formato de esquema para la conexión entre tablas.

Entre algunas limitaciones de esta implementación están:

- Los servidores remotos deben ser MySQL.
- No soporta transacciones.
- No soporta índices.
- No soporta ALTER TABLE
- Los BULK INSERT son lentos.
- No soporta *cache* de consultas.

² <http://dev.mysql.com/doc/refman/5.0/en/federated-storage-engine.html>

8. Federado vs. Clustered

Como se notó, en la sección de implementaciones no figura ninguna solución de Oracle, esto debido a una diferencia de arquitectura.

Oracle utiliza la arquitectura de clúster de disco compartido y no la de bases de datos federadas.

La arquitectura de clúster de disco compartido [12] está comprendida de servidores, de un clúster interconectado y de un subsistema de disco compartido. Una instancia de la base de datos se ejecuta en cada nodo, las transacciones se ejecutan en cada instancia que puede leer y actualizar cualquier parte de la base de datos. Esta arquitectura es implementada en la solución llamada *Oracle Real Application Cluster*.

En general, la comparación [12] se basa en que Oracle RAC maneja mejor las aplicaciones OLTP, y que los sistemas federados tienen deficiencias con respecto al desarrollo de aplicaciones, escalabilidad, disponibilidad y administración, que exactamente son las características que inicialmente no se garantizan en cierto grado para sistemas federados.

9. Conclusiones

Con base a la investigación documental anteriormente descrita, se puede concluir que los sistemas de bases de datos federadas son colecciones de componentes o nodos de múltiples bases de datos que cooperan entre sí, a través de un conjunto de esquemas de exportación y manteniendo su propia autonomía formando así una federación.

Las principales características son la heterogeneidad, que permite la existencia e interacción de diferentes sistemas y la autonomía, que se define como la capacidad de manejar su base de datos local, siendo así independiente.

Acerca de la arquitectura, se extiende la bien conocida ANSI/SPARC en esquemas de exportación y federados, con ciertas variaciones, como esquemas locales y de componentes. Además, se encontró una amplia variedad de modelos de datos mencionados en el documento.

El funcionamiento se divide en tareas de desarrollo, que tiene que ver con la forma interna de ejecución y en tareas de operación, que tiene que ver con la interacción de usuarios y sistemas.

Y finalmente, se presentaron algunas de las principales implementaciones, sobresalen la de SQL Server con su partición horizontal, la de IBM *Federated Database* con la propuesta de varios productos, el motor federado de MySQL que posee ciertas limitaciones y el enfoque diferido de clúster de disco compartido de Oracle.

10. Referencias

- [1] A. P. Sheth and J. A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," in *ACM Computing Surveys*, vol. 22, no. 3, United States of America, 1990, pp. 183-236.

- [2] R. J. Rabelo, H. Afsarmanesh, and L. M. Camarinha-Matos, "Applying Federated Databases to Inter-Organizational Multi-Agent Scheduling," in *1st International IFAC Workshop on Multi-Agent Systems in Production*, Vienna, Austria, 1999.
- [3] D. Heimbigner and D. McLeod, "A Federated Architecture for Information Management," in *ACM Transactions on Office Information System*, vol. 3, no. 3, United States of America, 1985, pp. 253-278.
- [4] P. Lim, S. Hwang, J. Srivastava, D. Clements, and M. Ganesh, "Myriad: Design and Implementation of a Federated Database Prototype," in *Software-Practice and Experience*, vol. 25, no. 2, United States of America, 1995, pp. 533-562.
- [5] G. Aslan and D. McLeod, "Semantic heterogeneity resolution in federated databases by metadata implantation and stepwise evolution," in *The VLDB Journal*, Springer-Verlag, 1999, pp. 120-132.
- [6] C. Bachman, "Summary of current work ANSI/X3/SPARC/study group: database systems," in *ACM SIGMOD Record*, vol. 6, United States of America, 1974, pp. 16-39.
- [7] C. Odoñez, Z. Chen, and J. García-García, "Metadata Management for Federated Databases," in *CIMS'07*, Lisboa, Portugal, 2007, pp. 31-38.
- [8] S. Hwang, J. Huang, and J. Srivastava, "Concurrency Control in Federated Databases: A Dynamic Approach," in *CIKM '93*, United States of America, 1993, pp. 694-703.
- [9] D. Fang, J. Hammer, and D. McLeod, "An Approach to Behavior Sharing in Federated Database Systems," University of Southern California, 1992.
- [10] H. Afsarmanesh, M. Wiedijk, and L. Hertzberger, "Flexible and Dynamic Integration of Multiple Information Bases," in *Proceedings DEXA'94 - 5th IEEE International Conference on Databases and Expert Systems Applications*, Springer-Verlag, 1994, pp. 277-288.
- [11] L. Haas and E. Lin, "IBM Federated Database Technology," IBM Corporation, 2002.
- [12] V. Buch, "Database Architecture: Federated vs. Clustered," Oracle Corporation, 2002.