Metodología para la BDD

Arturo Chinchilla S, Ronny Quesada A, Esteban Herrera V, Andrés Brais C
mchinchilla11@gmail.com, ronnyquesada96@gmail.com, nabetse.hv.09@gmail.com,
abcbrais@gmail.com
Área Académica de Administración de Tecnología de Información
Instituto Tecnológico de Costa Rica

***Resumen*—Este documento trata sobre la metodología utilizada para el diseño y generación de un sistema distribuido para la empresa courierTEC. Para este proyecto se utiliza Microsoft SQL Server como Sistema Administrador de Bases de Datos Distribuidas. Además, se hace un resumen de las funcionalidades esperadas por la empresa, la investigación realizada para la implementación del sistema y un ejemplo de dicha implementación en el contexto de este proyecto. El enfoque del diseño y la implementación se da sobre tres dimensiones sugeridas por [1], nivel de intercambio, Comportamiento de los patrones de acceso y nivel de conocimiento sobre el comportamiento del patrón de acceso.**

***Palabras clave*—Base de datos distribuida, SADB, SQL Server**

I. Introducción

El cantidad de información que manejan las organizaciones en la actualidad esta aumentando exponencialmente, por lo que los sistemas administradores de datos han evolucionado en busca de nuevos mecanismos para mejorar el control de datos a gran escala. Una de la tecnologías que trabaja en la dispersión de datos en sitios local o geográficamente dispersos y que resuelve los problemas de integración de la información son las Bases de Datos Distribuidas (BDD).

Las Bases de Datos Distribuidas representan la unión de tecnología de redes de computadoras y sistemas de bases de datos. [1] establecen que las Bases de Datos Distribuidas ”son una colección de bases de datos interrelaciones distribuidas sobre una red de computadoras”

Este documento explica la metodología utilizada para el diseño e implementación de una Base de Datos Distribuida con el objetivo de manejar un sistema de almacenaje y entrega de diferentes tipos de productos para la compañía courierTEC. Se utiliza esta tecnología dado la estructura de la compañía pues esta tiene 3 sucursales por lo que es conveniente el manejo distribuido de la información.

II. Descripción del problema

Para este proyecto se plantea la necesidad de la empresa courierTEC, la cual se encuentra iniciando servicios en el campo del almacenaje y distribución de paquetes para sus clientes, dicha necesidad radica en un sistema/servicio electrónico que permita registrar toda la información necesaria para que las operaciones de la empresa sean exitosas, como parte de la información proporcionada por la empresa se tiene:

II-A. Funcionalidades esperadas

Ya que la empresa cuenta actualmente con tres sucursales (Heredia, San José y Cartago), se requiere la implementación de un sistema distribuido, en el que se puedan registrar los servicios que utilizan los clientes en cada sucursal, así como la posibilidad de generar y monitorear diversos indicadores en tiempo real, esto para conocer el estado actual de cada sucursal en cualquier momento.

En la sucursal de Heredia se ubican las oficinas administrativas, por lo que es considerada la sucursal principal o central. Cada sucursal tiene su propio administrador, que a su vez tiene que reportar los resultados obtenidos al gerente en la sucursal central.

Se requiere una alta disponibilidad del sistema distribuido, de modo que si uno de los nodos (el sistema de una de las sucursales) sale de la red o presenta algún fallo, aún así se puedan consultar sus datos hasta el momento de la falla desde las oficinas centrales.

Para mayor facilidad en el manejo los paquetes, estos son clasificados mediante el uso de un tipo de paquete, que a su vez pertenece a una categoría específica.

Algunas de las consultas más frecuentes realizadas en las sucursales de San José y Cartago son:

- Cantidad de dinero recaudado por la sucursal.
- Cantidad de paquetes gestionados para un cliente en un rango de fechas definidas.
- Monto promedio pagado por cliente de los paquetes gestionados por la empresa para un rango de fechas definidas.
- Monto promedio pagado por todos los clientes para un tipo de paquete.

Además, en las oficinas centrales, el gerente realiza entre otras las siguientes consultas:

- Monto recaudado por sucursal para un período específico.
- Monto recaudado por sucursal para un tipo de paquete específico en un período específico.
- Listado de los tres mejores clientes (los que tengan un monto mayor en el total de paquetes que hayan traído).

III. Investigación

El diseño de un sistema informático distribuido implica tomar decisiones sobre la ubicación de datos y programas en los sitios de una red informática, así como posiblemente diseñar la red en sí misma. En el caso de DBMS distribuidos, la distribución de las aplicaciones implica dos cosas: la distribución del software DBMS distribuido y la distribución de los programas de aplicación que se ejecutan en él [1]. Dado que en cursos anteriores de bases de datos el equipo de trabajo utilizo Microsoft SQL Server, para este proyecto se utiliza este DBMS dado a su gran capacidad operativa en BDD y ademas por la facilidad que ofrece el Tecnológico de Costa Rica para la descarga de este software en su versión Enterprise.

Se ha sugerido que la organización de sistemas distribuidos puede investigarse a lo largo de tres dimensiones ortogonales [Levin y Morgan, 1975] [1].

1. Nivel de intercambio
2. Comportamiento de los patrones de acceso
3. Nivel de conocimiento sobre el comportamiento del patrón de acceso

En términos del nivel de intercambio, hay tres posibilidades. Primero, no hay intercambio: cada aplicación y sus datos se ejecutan en un sitio, y no hay comunicación con ningún otro programa ni acceso a ningún archivo de datos en otros sitios. Esto caracteriza los primeros días de la creación de redes y probablemente hoy no sea muy común. Luego encontramos el nivel de intercambio de datos; todos los programas se replican en todos los sitios, pero los archivos de datos no. En consecuencia, las solicitudes de los usuarios se manejan en el sitio donde se originan y los archivos de datos necesarios se mueven por la red. Finalmente, en el intercambio de datos más programas, tanto los datos como los programas pueden compartirse, lo que significa que un programa en un sitio determinado puede solicitar un servicio de otro programa en un segundo sitio, que a su vez puede tener que acceder a un archivo de datos ubicado en un tercer sitio [1].

Para este proyecto se utiliza el Nivel de Intercambio del Datos, puesto que las solicitudes que realizan los administradores y clientes de cada sucursal se manejan en su respectivo sitio y los archivos de datos necesarios se mueven por la red mediante replicación.

En la segunda dimensión(comportamiento del patrón de acceso), es posible identificar dos alternativas. Los patrones de acceso de las solicitudes de los usuarios pueden ser estáticos, de modo que no cambien con el tiempo o ser dinámicos [1].

Para este proyecto se utiliza los patrones de acceso estáticos, a través de los parámetros que recibe los procedimientos almacenados se establece la estructura de acceso a la información y estos no va a cambiar con el tiempo.

La tercera dimensión de la clasificación es el nivel de conocimiento sobre el comportamiento del patrón de acceso. Una posibilidad, por supuesto, es que los diseñadores no tengan ninguna información sobre cómo los usuarios accederán a la base de datos. Esta es una posibilidad teórica, pero es muy difícil, si no imposible, diseñar un SGBD distribuido que pueda hacer frente con eficacia a esta situación. Las alternativas más prácticas son que los diseñadores tengan información completa, donde los patrones de acceso puedan predecirse razonablemente y no se desvíen significativamente de estas predicciones, o información parcial, donde haya desviaciones de las predicciones [1].

En caso de proyecto se tiene la información completa, dado que se cuenta con un documento de requerimientos proveídos por la profesora.

Dos estrategias principales que se han identificado para diseñar bases de datos distribuidas son el enfoque top-down y el enfoque Bottom-up [Ceri et al., 1987] [1].

Enfoque TOP-DOWN [1]

- Inicia con el análisis de los requerimientos los cuales define el ambiente del sistema.
- Se realiza el diseño de las vistas(interfaces de usuarios),
- Se crea el diseño conceptual, el cual puede dividirse en análisis de entidades(relaciones, atributos) y en análisis funcional del sistema.
- El usuario provee información estadística como la frecuencia de uso de las aplicaciones y el volumen de información.
- El siguiente paso es el diseño de la distribucion, se distribuyen las entidades en los sitios del sistema y para cada sitio se realiza un esquema local. Consiste en dos pasos fragmentación y asignación.
- El ultimo paso es el diseño físico, en el cual se mapea el diseño conceptual local a los dispositivos de almacenamiento físico.

Enfoque BOTTOM-UP [1]

- Este enfoque es conveniente cuando las bases de datos ya existen y las tareas del proceso de diseño involucran integrarlas en una sola base de datos.
- El punto de inicio de este proceso son los esquemas conceptuales locales individuales;

el proceso consiste en integrar esos esquemas en esquema conceptual global.

Es evidente que para la implementación del proyecto no se tenia ninguna base de datos, por lo que equipo de trabajo opta por el enfoque TOP-DOWN.

Esencialmente las instancias de relación son tablas y una de las situaciones es encontrar un método o forma de convertir la tabla completa en sub-tablas mas pequeñas. Hay dos alternativas para esto: Fragmentación horizontal(se da sobre las tuplas) y fragmentación vertical(se da sobre los atributos)[1]

En el diseño de la BDD de courierTEC se utiliza solamente fragmentación horizontal, pues solo es necesario información de las tuplas de determinada tabla.

Una vez fragmentada la base de datos se procede a su asignación en los diferentes sitios de la red, esto se realiza mediante replicación por su eficiencia en las lecturas y la posibilidad de ejecutar en paralelo, sin embargo el argumento mas importante es por que uno de los requerimientos establece que se requiere una alta disponibilidad del sistema, de forma tal que si uno de los nodos (una sucursal) sale de la red, aún así se puedan consultar sus datos desde las oficinas centrales. Esta replicación solamente se da de manera unidireccional desde los nodos de las sucursales de San José y Cartago hacia el nodo Central, y solamente para ciertas tablas, donde sus datos son estrictamente necesarios para las operaciones realizadas por el gerente en las oficinas centrales.

IV. Ejemplo de la implementación

En la figura 1 se muestra un ejemplo de la arquitectura implementada para la Base de Datos Distribuida implementada en este proyecto, donde existen tres nodos, dos de los cuales (Cartago y San José) se conectan con las sucursal de Heredia para ejecutar transacciones.

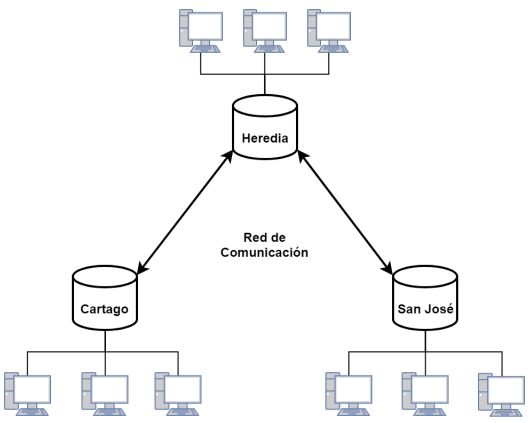

Figura 1. Implementación de la base de datos distribuida

V. Conclusiones

Ya que no existía un sistema u otras bases de datos sobre las cuales trabajar, el enfoque TOP-DOWN fue el más adecuado de seguir. Además el documento facilitado por la empresa, que presentaba información sobre las principales consultas realizadas en cada sucursal fue de vital importancia para la implementación del Sistema, la fragmentación y la asignación de datos, de esta manera se logra optimizar las consultas y el uso de recursos como tráfico en la red, procesamiento y almacenamiento de datos. También ayudó en el diseño de la Base de Datos, ya que ofrecía detalles sobre atributos de las relaciones que se deseaban registrar.

SQL Server en su versión Enterprise, al ser un SGBD robusto y ya conocido de otros cursos, que permite la creación, administración y manipulación de bases de datos distribuidas, hizo que la implementación de la base de datos no presentó dificultad alguna, además, ya que la empresa courierTEC esperaba que el sistema presentara alta disponibilidad, de manera que la Sucursal Central pudiera realizar las consultas aunque un nodo saliera de la red, la replicación de los datos presentada por SQL Server vino a llenar dicha solicitud.

Para cada nodo existe autonomía local, de modo que no dependen de otros nodos para su correcto funcionamiento, esto en caso de que alguno por algún motivo deje de funcionar.

Referencias

[1] M. Özsu and P. *Valduriez, Principles of distributed database systems*, 3rd ed. New York (NY): Springer, 2011.