Ambiente de experimentación para Bases de Datos Distribuidas

Lic. Rodolfo Bertone[1], MS Jorge Ardenghi[2], Ing. Armando De Giusti[3]

UNLP-UNS

Introducción

Una Base de Datos Distribuidas (BDD) puede ser definida como una colección integrada de datos compartidos que están físicamente repartido a lo largo de los nodos de una red de computadoras. Un DDBMS es el software necesario para manejar una BDD de manera que sea transparente para el usuario. [BURL 94]

Un DBMS centralizado es un sistema que maneja una BD simple, mientras que un DDBMS es un DBMS simple que maneja múltiples BD. El término global y local se utiliza, cuando se discute sobre DDBMS, para distinguir entre aspectos que se refieren al sitio simple (local) y aquello que se refiere al sistema como un todo (global). La BD local se refiere a la BD almacenada en un sitio de la red, mientras que la DB global se refiere a la integración lógica de todas las BD locales. [BELL 92]

El propósito de la BD es integrar y manejar los datos de una empresa o corporación. La motivación para definir esta BD es la de tener la información relevante de las operaciones de la empresa en un único almacenamiento, de manera que todos los problemas asociados con aplicaciones en una empresa puedan ser servidos de una manera uniforme. La dispersión geográfica de estas empresas hace que un modelo centralizado no aplicable, siendo un modelo distribuido la solución apropiada. [SCHU 94] [BHAS 92]

El modelo distribuido de datos hace posible la integración de BD heterogéneas proveyendo una independencia global del DBMS respecto de esquema conceptual. Además, es posible implementar una integración tal, que reúna varios modelos de datos, representado cada uno de ellos características propias de empresas diferentes, asociadas para un trabajo conjunto. Este modelo de distribución, genera las denominadas Bases de Datos Federativas. [LARS 95] [SHET 90]

Con los modelos distribuidos o federativos de datos, surgen una serie de consideraciones especiales y problemas potenciales que no estaban presente en los sistemas centralizados. La replicación y fragmentación de información hace a que los datos estén más "cerca" y "disponibles" para los usuarios. Estos dos conceptos hacen que la preservación de la integridad y consistencia de la información deba tratarse con nuevos mecanismos que contemplen la ubicación de los datos y las copias existentes de los mismos. [DATE 93] [THOM 90]

Dos de los conceptos más interesantes, para su estudio y evaluación, relacionados con las BDD son:

- Replicación de la información: repetición del mismo dato o juego de datos en más de un nodo de la red de computadoras que forman el sistema distribuido. El grado de replicación determina el grado de disponibilidad de la información, evitando el problema de los sistemas centralizados, donde en caso de caídas de dicho nodo, no se podía acceder a la información. Las desventajas que presenta la replicación están asociadas con las dificultades para el mantenimiento de la integridad

[1] Profesor Adjunto Dedicación Exclusiva, E-mail: pbertone@lidi.info.unlp.edu.ar
[2] Profesor Titutar Dedicación Exclusiva, Depto. Ciencia de la Computación UNS.
E-mail: jrs@criba.edu.ar
[3] Profesor Titular Dedicación Exclusiva, Investigador Principal del CONICET. UNLP.
E-mail: degiusti@lidi.info.unlp.edu.ar

y consistencia de información. Esto es, se tiene el mismo dato varias veces, una actualización debe realizarse sobre todas las copias.

- Fragmentación de la información: al tener varios nodos que actúan en forma independiente es posible (y necesario) colocar en diferentes datos en diferentes lugares. De esta forma se logra que la información se encuentre "cerca" del usuario. Las decisiones de cómo efectuar la fragmentación y como la misma afecta la perfomance del sistema es un aspecto muy interesante en el estudio del comportamiento de las BDD.

La posibilidad de replicación y fragmentación de la información en un entorno distribuido hace que los procesos para el control de concurrencia de sistemas centralizados sufran alteraciones para adaptarse a las nuevas necesidades. La solución más sencilla consiste en tener un gestor de bloqueos centralizado y canalizar todos los pedidos de información en él. Esta solución genera un punto único de fallos y un cuello de botella, ambas situaciones no son deseadas en un entorno distribuido. Existen varias alternativas distribuidas como solución:

- Protocolos de mayoría: ante un pedido de dato, se solicita acceso a la información a todos los nodos, si la mitad más uno responde afirmativamente se obtiene el uso exclusivo.
- Protocolos de preferencia: más útil en caso de tener más consultas que actualizaciones. Ante un lock compartido, si el dato puede obtenerse desde un nodo se le asigna a la localidad que lo solicita. Esto hace que ante un lock exclusivo deba tenerse la certeza de acceso único al elemento.

De la misma forma que los protocolos para concurrencia sufren modificaciones para un entorno distribuido, con los protocolos para el mantenimiento de integridad de los datos ocurre un caso similar. Si bien cada localidad es la encargada de garantizar la integridad de su información, todas las localidades deben cooperar puesto que una transacción puede estar accediendo a datos ubicados en más de un nodo. Además, pueden surgir otros problemas: fallo de una localidad, fallo en la comunicación, bloqueo, etc. Las soluciones presentadas para salvar estos nuevos inconvenientes son:

- Protocolo de dos fases
- Protocolo de tres fases

Dentro del ambiente de simulación está previsto estudiar, implementar y comparar ambos protocolos.

Objetivo del Proyecto

El objetivo fundamental es definir, modelar e implementar un ambiente de simulación para el mantenimiento y recuperación de datos, en un sistema de BDD, donde se puedan estudiar, monitorear, medir y posteriormente comparar los resultados obtenidos, de la ejecución de transacciones distribuidas a partir de las características de replicación y fragmentación de datos en un entorno distribuidos.

Otro aspecto importante dentro de los objetivo es generar de manera aleatoria y parametrizable diferentes las trazas de ejecución de transacciones, representando cada una de ellas entornos diferentes de distribución de información.

Este generador de casos de prueba, permite determinar la cantidad de tablas que componen la BDD, y la cantidad de transacciones involucradas en la simulación como parámetros iniciales. El segundo grupo de parámetros permite identificar el tipo de simulación, definiendo para ello el porcentaje de replicación de los datos, desde un modelo distribuido sin replicación, hasta un nivel de replicación de 100% de los datos.

Qué significa que los datos estén replicados un 100%? Básicamente puede significar:

1) que cada información contenida en la BDD está, al menos dos veces;
2) que algunos (o todos) los datos se encuentre disponibles en todos los nodos de la red.

Como aspecto de simulación es interesante evaluar las dos alternativas, si bien hay que tener en cuenta que una distribución total de los datos en todos los nodos de la red no representa una situación real. En una primer etapa, el nivel de replicación indica que porcentaje de los datos estará repetido sin importar el número de nodos donde se localice la información.

Siguiendo con los parámetros del generador de trazas es posible definir el porcentaje de transacciones locales, sobre las transacciones globales. Si bien es consabido que la proporción estimada es 85 – 15, consideramos que la posibilidad de definición de esta variable permitirá estudiar comportamiento en sistemas que se alejen de las pautas tradicionales de trabajo.

Durante la introducción se mencionaron los protocolos de commit dos y tres fases como procesos para el control de la integridad y consistencia de la información, y quedó establecido las ventajas y desventajas del primero respecto del segundo. Nuestra propuesta considera interesante evaluar el comportamiento de los mismos, por lo tanto está previsto la ejecución de las trazas siguiendo las dos políticas establecidas.

Como política para el mantenimiento de integridad en la información se decidió utilizar aquella basada en bitácora, la cual según los estudios realizados, es más fácil y eficiente respecto de la conocida como doble paginación. Como es sabido esta técnica basada en bitácora tiene dos alternativas: 1) modificación inmediata de los datos, 2) modificación diferida. Las características de cada una de ellas hacen que la información que es necesaria guardar en el *log* sea diferente, el modelo de simulación permite establecer el tipo de política y, eventualmente, comparar los resultados obtenidos.

Por último, es posible definir el porcentaje o probabilidad de fallos que se presentará en la simulación.

La arquitectura del sistema de simulación está especificada y modelizada teniendo en cuenta los siguientes procesos que componen cada localidad:

- Simulación de cada nodo representando una localidad de la red.
- Cliente: simulando a la ejecución de la transacción en cada momento
- Gestor de Transacciones: es el proceso encargado de controlar la ejecución local de la transacción
- Coordinador de transacciones: es el encargado de controlar la ejecución, decidiendo si la transacción puede completarse localmente. En caso de no serlo, decide como dividir la transacción y a que otras localidades debe solicitar el servicio. Para ello cuenta con una tabla de estado, donde puede encontrar la ubicación de los datos y la disponibilidad (si la localidad se encuentra activa o a fallado). Esta tabla se mantiene dinámicamente.
- Gestor de bloqueos: proceso encargado de administrar los bloqueos en el acceso a los datos locales gerenciados por la localidad.
- Administrador de la bitácora.

Se ha discutido además, consideraciones respecto al diccionario de datos distribuidos (la tabla anteriormente mencionada). En una primer etapa se decidió concentrar esta información en un solo nodo de manera que fuera consultado por las demás localidades, a fin de facilitar la simulación. Como es sabido, esta solución genera un punto único de fallos, circunstancia no deseada en un entorno distribuido, por lo tanto el objetivo es distribuir esta información en cada localidad. Esto motiva que el mantenimiento de esta tabla sea dinámico y necesariamente distribuido.

Otro aspecto interesante de modelización y estudio de comportamiento, que se presenta cuando la replicación de los datos se establece en forma dinámica, analizando como afecta esta condición la perfomance de las consultas contra los ABM. Además, se puede evaluar los algoritmos de recuperación en dichas circunstancias. Esta simulación se ha evaluado pero aún está en una etapa de modelización. [WOLF 97]

El procesamiento distribuido se simula utilizando un soporte de PVM (Parallel Virtual Machine) [GEIS 94]. Este, provee un framework unificado con el cual se pueden desarrollar programas con

sistemas heterogéneos sobre hardware heterogéneo. En forma transparente al usuario se rutean los mensajes, transfieren los datos y se realiza el schedule de tareas a través de arquitecturas de redes que pueden ser diferentes.

Resultados obtenidos y esperados

En el proyecto se ha trabajado en la definición de la clase de problemas mencionado anteriormente modelizando el problema como un todo, para luego pasar a la implantación y obtención de resultados en partes. Se han tenido algunos resultados (publicaciones, Tesinas de Grado), y se está trabajando en la evaluación más detallada de nuevos casos. Entre las experiencias concretas realizadas puede mencionarse:

- Simulación de fallas en un entorno distribuido, con datos fragmentados sin replicación. [MIAT98]
- Estudio de la influencia en la proporción de acceso a datos locales y globales, sin replicación, con replicación parcial o total, en diversos casos de prueba.
- Estudios preliminares de los protocolos de dos y tres fases para recuperación de datos ante fallos, con diversos grados de replicación.
- Actualmente se está avanzando en la generación del ambiente de simulación para asegurar integridad y consistencia de información utilizando el protocolo de dos fases con fragmentación y replicación de información.

Bibliografía

[BELL 92] *Distributed Database Systems*, Bell, David; Grimson, Jane. Addison Wesley. 1992

[BHAS 92] *The architecture of a heterogeneous distributed database management system: the distributed access view integrated database (DAVID).* Bharat Bhasker; Csaba J. Egyhazy; Konstantinos P. Triantis. CSC '92. Proceedings of the 1992 ACM Computer Science 20th annual conference on Communications, pages 173-179

[BONT 95] *Dtabase management, Principles and productos.* Bontempo, Charles; Maro Saracco, Cynthia. Prentice Hall 1995.

[BURL 94] *Managin Distributed Databases. Building Bridges between Database Island.* Burleson, Donal. 1994

[DATE 93] *Introducción a los sistemas de Bases deDatos*. Date, C.J. Addison Wesley 1993.

[GEIS 94] *PVM: Parallel Virtual Machine. A User guide and tutorial for networked parallel comoputing.* Geist, Al; Beguelin, Adam; Dongarra, Jack; Jiang, Weicheng; Manchek, Robert; Sunderam, Vaidy. The MIT Press. 1994.

[LARS 95] *Database Directions. From relational to distributed, multimedia, and OO database Systems.* Larson, James. Prentice Hall. 1995

[LEUN 93] *High-performance parallel database architecture.* C. H. C. Leung; H. T. Ghogomu. Proceedings of the 1993 international conference on Supercomputing, pages 377-386.

[MIAT 98] *Experiencias en el análisis de fallas en BDD*. Miatón, Ivana; Ruscuni, Sebastián; Bertone, Rodolfo; De Giusti, Armando. Anales CACIC 98. Neuquén Argentina.

[SCHU 94] *The Database Factory. Active databse for enterprise computing.* Schur, Stephen. 1994.

[SHET 90] *Federated database systems for managing distributed, heterogeneous, and autonomous databases.* Amit P. Sheth; James A. Larson. ACM Computing Surveys.Vol. 22, No. 3 (Sept. 1990), Pages 183-236

[THOM 90] *Heterogeneous distributed database systems for production use.* Thomas, Charles; Glenn R. Thompson; Chin-Wan Chung; Edward Barkmeyer; Fred Carter; Marjorie Templeton; Stephen Fox; Berl Hartman. ACM Computing Surveys. Vol. 22, No. 3 (Sept. 1990), Pages 237-266

[WOLF 97] *An adaptive data replication algorithm.* Ouri Wolfson; Sushil Jajodia; Yixiu Huang; ACM Transactions on Database Systems. Vol. 22, No. 2 (June 1997), Pages 255-314