Productos y Tecnologías de Oracle TimesTen

Informe Ejecutivo de Oracle
Febrero de 2007

Productos y Tecnologías de Oracle TimesTen

Introducción..... 3
"Cuando los Milisegundos Cuentan"..... 3
El Crecimiento de Aplicaciones de Tiempo Real..... 4
Industrias en Tiempo Real..... 4
Empresas en Tiempo Real..... 4
Software para la Administración de Datos en Tiempo Real..... 4
Implementación en el Nivel de Aplicaciones..... 4
Productos..... 5
Oracle TimesTen In-Memory Database..... 5
Replicación - TimesTen a TimesTen..... 5
Cache Connect to Oracle..... 5
Tecnología In-Memory Database..... 5
Las Estructuras Físicas de Oracle TimesTen..... 6
Bibliotecas Compartidas en el Nivel de Aplicaciones..... 6
Estructuras de Datos que Residen en Memoria..... 7
Procesos del Sistema..... 7
Programas Administrativos..... 7
Puntos de Control y Archivos Testigo..... 7
Tecnología de Replicación de Datos..... 8
Tecnología Caché…..... 8
Tecnología IMDB Detallada..... 9
Optimización de Consultas..... 9
Administración de Grupos de Buffer..... 10
Estructuras de Índices..... 10
Por qué la Diferencia..... 11
Una Nueva Mirada hacia el Rendimiento de Precios..... 11
Desempeño Excepcional..... 11
Escalabilidad..... 11
Tiempo de Respuesta..... 12
Funcionalidad en Tiempo Real..... 12
Administración de Datos..... 12
Procesamiento de Consultas..... 16
Replicación de Datos..... 16
Caché..... 17
Procesamiento de Eventos..... 19
Conclusión..... 20

Productos y Tecnologías de Oracle TimesTen

INTRODUCCIÓN

Oracle TimesTen In-Memory Database es una base de datos relacional de memoria optimizada que ofrece tiempos de respuesta muy bajos y alto rendimiento para sus sistemas de desempeño crítico. Apunta a ejecutarse en el nivel de aplicaciones, cerca de las aplicaciones y, opcionalmente, en proceso con las aplicaciones. Puede utilizarse como la base de datos de archivo, o como caché de Oracle Database.

Este informe describe los productos y tecnologías de Oracle TimesTen y los puntos de integración con otros productos de Oracle. Contribuye con una evaluación "escrita" antes de una experiencia práctica con el software y los manuales.

"Cuando los Milisegundos Cuentan"

Los productos de Oracle TimesTen ofrecen administración de datos en el nivel de aplicaciones para los sistemas de desempeño crítico, optimizados para brindar tiempos de respuesta extremadamente rápidos y caché de datos Oracle en tiempo real. Con Oracle TimesTen, las empresas extienden sus infraestructuras de software para crear sistemas con:

• Capacidad de respuesta instantánea
• Alta escalabilidad
• Disponibilidad continua

Estos sistemas se utilizan para:

• Aumentar la fidelidad de los clientes
• Atraer a nuevos clientes
• Optimizar las operaciones
• Evitar la costosa alternativa de desarrollo de un software de propiedad.

TimesTen cuenta con un probado historial de implementaciones de producción desde 1998 en empresas que trabajan en tiempo real y en sectores donde el tiempo es un factor crítico (servicios de telecomunicaciones por red, sistemas de soporte operativo, centros de contacto, sistemas de reservas y líneas aéreas, sistemas de comandos y control, y comercialización de títulos valores). Cientos de empresas en todo el mundo utilizan Oracle TimesTen en sus aplicaciones de producción, con inclusión de Amdocs, Aspect, Avaya, el Mercado de Valores de Bombay, Cisco, Ericsson, JP Morgan, Lucent, NEC, Nokia, Salesforce.com y Sprint.

Los productos de Oracle TimesTen ofrecen administración de datos en el nivel de aplicaciones para los sistemas de desempeño crítico, optimizados para brindar tiempos de respuesta extremadamente rápidos y caché de datos Oracle en tiempo real.

EL CRECIMIENTO DE APLICACIONES EN TIEMPO REAL

Industrias en Tiempo Real

Para muchas empresas, las aplicaciones en tiempo real no son optativas. Son una necesidad para formar parte del mercado. Los fabricantes de equipos de red, los operadores de telecomunicaciones, los agentes de la bolsa de valores, las líneas aéreas, las empresas marítimas y de logística, y las agencias de defensa e inteligencia son los principales ejemplos. Anteriormente, crear aplicaciones en tiempo real para estas industrias también requería el desarrollo de un software de infraestructura en tiempo real. No existían opciones comerciales. Rápidos pero inflexibles, estos sistemas hacían el trabajo, siempre y cuando las aplicaciones no cambiasen sus requerimientos. Pero los sectores dinámicos pronto dejarían atrás las aplicaciones estáticas. El costo de desarrollar, probar y mantener un software de infraestructura especializado raramente se justifica cuando existen alternativas comerciales.

Empresas en Tiempo Real

Con la creciente velocidad de los mensajes que se transmiten a través de redes comerciales, el uso de procesos en tiempo real para capturar, analizar y responder inteligentemente a los eventos clave se está convirtiendo en la prueba de desempeño para alcanzar la excelencia corporativa. Esto no solo es importante para la ejecución y administración de los procesos comerciales críticos. Los clientes esperan interacciones altamente personalizadas y la máxima capacidad de respuesta de cualquier empresa con la que hagan negocios.

El monitoreo de la actividad comercial, el procesamiento de eventos complejos, las aplicaciones basadas en RFID/sensores, los portales Web y los Servicios Web contribuyen al movimiento de las aplicaciones más allá de los límites de la empresa. Configuradas como una recopilación dinámica de rutinas interrelacionadas, estas aplicaciones son parte de un enfoque global conocido como "arquitectura orientada a servicios" (SOA). A pesar del movimiento de muchas aplicaciones hacia los límites de la empresa, la mayoría de las fuentes de datos aún reside en el *back office*, dominadas por grandes cantidades de datos antiguos casi intactos que rodean cantidades más pequeñas de información actualmente activa. Una extensión natural de los conceptos SOA incluye la administración ligera y en tiempo real de datos en el nivel de aplicaciones, conectada con las fuentes de datos corporativos a fin de brindar desempeño en tiempo real para los datos actualmente activos.

Software para la Administración de Datos en Tiempo Real

Las arquitecturas empresariales cuyo mayor beneficio proviene del procesamiento en tiempo real ofrecen administración de transacciones, datos y eventos en el nivel de aplicaciones, potenciando los sistemas operativos con respuestas rápidas y conocimientos más detallados. El simple hecho de recopilar datos y colocarlos en caché junto a las aplicaciones no es suficiente, como a menudo ocurre con los proyectos internos de primera generación. Ni tampoco es práctico situar la base de datos corporativa en la misma plataforma que la de las aplicaciones.

Lo que se necesita es la generación de software de infraestructura liviana con interfaces conocidas y avanzadas, y lenguajes de consulta ampliamente utilizados –que puedan interconectarse fácilmente con bases de datos de back-office existentes, sistemas de mensajería y servidores de aplicaciones– y que explote todo el potencial de desempeño de las actuales plataformas informáticas con mucha memoria y conectadas en red. Esta es la generación de software de infraestructura para la administración de datos en tiempo real provista por Oracle TimesTen.

Los fabricantes de equipos en red, los operadores de telecomunicaciones, los agentes de la bolsa de valores, las líneas aéreas, las empresas marítimas y de logística, y las agencias de defensa e inteligencia son los principales ejemplos de los sectores donde las aplicaciones en tiempo real constituyen una necesidad

El uso de procesos en tiempo real para capturar, analizar y responder inteligentemente a los eventos clave se está convirtiendo en la prueba de desempeño para alcanzar la excelencia corporativa. El simple hecho de recopilar datos y colocarlos en caché junto a las aplicaciones no es suficiente, ni tampoco es práctico situar la base de datos corporativa en la misma plataforma que la de las aplicaciones.

Implementación en el Nivel de Aplicaciones

Gran parte del nuevo desarrollo de aplicaciones de hoy se focaliza en mejorar las interacciones con los clientes y optimizar las operaciones internas para eliminar las demoras y los costos excesivos. Estas son aplicaciones en tiempo real que residen cerca del límite de la red o, en algunos casos, dentro de la red como servicios *hosted*.

Este es el nuevo nivel de aplicaciones empresariales, con diferentes requisitos de disponibilidad, desempeño y plataforma en relación con las aplicaciones de back-office tradicionales. Los eventos de negocio están incluidos en los mensajes de red a los cuales se suscriben las aplicaciones, lo cual activa el procesamiento en tiempo real y la publicación de mensajes adicionales.

Para cumplir con los objetivos de escalabilidad y tiempo de respuesta de estas aplicaciones, a menudo es necesario implementar el software de infraestructura y las aplicaciones que coexisten en la misma plataforma, con inclusión de algunos o todos los datos que controlan la aplicación. Los productos de Oracle TimesTen han sido diseñados para integrarse fácilmente en estos entornos, con una arquitectura

optimizada para la implementación en el nivel de aplicaciones y opciones de configuración para activar soluciones altamente personalizadas.

PRODUCTOS

El software de administración de datos en tiempo real de Oracle TimesTen consta de tres productos basados en una base de datos *in-memory*, replicación de datos y tecnologías en caché. Esta parte del informe brinda una introducción general a estos productos y tecnologías. Las siguientes secciones presentan información detallada adicional.

Oracle TimesTen In-Memory Database

Oracle TimesTen In-Memory Database es una base de datos relacional, con memoria optimizada, que otorga a las aplicaciones una capacidad de respuesta instantánea y un rendimiento de alto desempeño, algo muy requerido por las actuales empresas y sectores en tiempo real, como el sector de telecomunicaciones, mercado de capitales y defensa. Implementado en el nivel de aplicaciones como base de datos incorporada o en caché, Oracle TimesTen In-Memory Database opera en las bases de datos que se adaptan completamente a la memoria física utilizando interfaces SQL estándar.

Replicación – *TimesTen* a *TimesTen*

Replicación – TimesTen a TimesTen es una opción de Oracle TimesTen In-Memory Database que permite la replicación de datos en tiempo real entre los servidores, a fin de compartir las cargas y obtener una alta disponibilidad. Las configuraciones de replicación de datos pueden ser *activa-standby* o *activa-activa*, utilizando una transmisión sincronizada o asíncrona, con detección y resolución de conflictos y resincronización automática posterior a la restauración de un servidor que ha fallado. La replicación de datos es completamente compatible con la opción *Cache Connect to Oracle*.

Lo que se necesita es la generación de software de infraestructura liviana que explote todo el potencial de desempeño de las actuales plataformas informáticas con mucha memoria y conectadas en red. Los productos de Oracle TimesTen se integran fácilmente en estos entornos, con una arquitectura optimizada para su implementación en el nivel de aplicaciones.

Oracle TimesTen consta de tres productos basados en la base de datos *in-memory*, la replicación de datos y las tecnologías en caché.

Cache Connect to *Oracle*

Cache Connect to Oracle es una opción de Oracle TimesTen In-Memory Database que crea un caché actualizable, en tiempo real, para los datos de Oracle que residen en el nivel de aplicaciones. Descarga ciclos informáticos de sistemas de back-end y permite utilizar aplicaciones escalables, en tiempo real y con una notable capacidad de respuesta. *Cache Connect to Oracle* carga un subgrupo de datos Oracle a TimesTen, propaga las actualizaciones en ambas direcciones, automatiza la transferencia de solicitudes SQL para los datos no colocados en caché y vuelve a sincronizar automáticamente los datos después de las fallas. *Cache Connect to Oracle* es totalmente compatible con la opción Replicación –TimesTen a TimesTen.

TECNOLOGÍA IN-MEMORY DATABASE

La tecnología in-memory database (IMDB) es la tecnología fundamental para los productos Oracle TimesTen. La tecnología IMDB implementa una base de datos relacional en donde todos los datos que se están ejecutando residen en RAM, y las estructuras de datos y los algoritmos de acceso explotan esta propiedad para alcanzar un desempeño extraordinario. Comparada con un sistema RDBMS totalmente en caché, la tecnología IMDB utiliza menos espacio de CPU, debido a que se eliminan los gastos generales de administrar los buffers de memoria y rendir cuenta de la ubicación de múltiples datos (disco y memoria). Con la tecnología IMDB, se utilizan discos magnéticos para fines de persistencia y recuperación, más que como ubicación principal de almacenamiento para la base de datos.

El desempeño de Oracle TimesTen con memoria optimizada se complementa con la funcionalidad que otorga propiedades transaccionales, mecanismos de persistencia y recuperación ante las fallas del sistema. Hay una variedad de alternativas disponibles para las funciones de bloqueo, aislamiento de múltiples usuarios e ingreso al sistema, con una gran variedad de escenarios de aplicaciones, desde cachés de consulta transitorios hasta los principales sistemas de facturación y negociación de transacciones.

En TimesTen, la durabilidad es alcanzada al registrar en el disco los cambios de las transacciones validadas y actualizar periódicamente la imagen de un disco de la base de datos, lo cual es denominado "punto de control". El tiempo de escritura en disco para el registro puede ser configurado por la aplicación, ya sea sincrónicamente al finalizar la transacción, o de manera diferida hasta más adelante, dando como resultado un mejor desempeño. Muchas situaciones favorecen la obtención de un mejor desempeño para lograr un registro sincronizado, particularmente cuando el valor monetario de una transacción es bajo o los datos de transacción son efímeros, como cuando se rastrea la ubicación de celulares móviles en una red que comunican la ubicación de su equipo cada pocos segundos.

Las interfaces nativas respaldadas por los productos de Oracle TimesTen son compatibles con los estándares y, por lo general, también son compatibles con cualquier otra base de datos relacional que se ajuste a los estándares. Las aplicaciones emiten comandos SQL (lenguaje de consulta estructurado) a través de JDBC (conectividad de base de datos Java) o interfaces ODBC (conectividad de base de datos abierta). Las sentencias para definir las bases de datos, las configuraciones de replicación y los grupos de caché también se adhieren a las convenciones de sintaxis de SQL. SNMP (protocolo de administración de red simple) es utilizado para emitir alertas estandarizadas para la administración del sistema.

Una API abierta para el Registro de Transacciones (*Transaction Log API* - XLA) con una interfase JMS estándar es proporcionada para leer el registro de la transacción. Esto es útil para crear aplicaciones que reaccionan ante las actualizaciones de base de datos. En este sentido, XLA es un "trigger" liviano. También es una manera de generar replicación de datos personalizada de Oracle TimesTen a otros sistemas de base de datos.

La tecnología in-memory database implementa una base de datos relacional donde todos los datos que se están ejecutando residen en RAM, y las estructuras de datos y los algoritmos de acceso explotan esta propiedad para lograr un desempeño extraordinario.

El desempeño de Oracle TimesTen se complementa con una funcionalidad que otorga propiedades transaccionales, mecanismos de persistencia y recuperación ante las fallas del sistema.

LAS ESTRUCTURAS FÍSICAS DE ORACLE TIMESTEN

Esta sección describe los componentes del sistema de los productos Oracle TimesTen, desde la perspectiva de qué se instala, y qué ejecuta y consume los recursos informáticos.

Los productos de Oracle TimesTen son una combinación de:
• Bibliotecas compartidas
• Estructuras de datos que residen en la memoria
• Procesos del sistema
• Programas administrativos
• Puntos de control y archivos testigo en disco

Bibliotecas Compartidas en el Nivel de Aplicaciones

Las rutinas que realizan operaciones SQL y funciones relacionadas se incorporan en un grupo de bibliotecas compartidas que los desarrolladores vinculan con sus aplicaciones y ejecutan como parte del proceso de aplicaciones. Este enfoque de biblioteca compartida se contrapone a un RDBMS más convencional, el cual es implementado como una recopilación de programas ejecutables a los cuales se conectan las aplicaciones, generalmente a través de una red cliente/servidor. Normalmente, incorporar bibliotecas de administración de datos en la aplicación podría hacer que la base de datos se vuelva vulnerable a posibles corrupciones si el proceso de aplicaciones finaliza de manera anormal. Oracle TimesTen ha resuelto este desafío. Al utilizar un algoritmo con patentes pendientes, conocido como "MicroLogging", las bibliotecas de Oracle TimesTen se autoprotegen de las fallas en el proceso de aplicaciones. Las bases de datos *in-memory* mantienen su consistencia, y las otras aplicaciones siguen funcionando sin ningún impacto.

Figura 1. Componentes del producto Oracle TimesTen In-Memory Database

Las aplicaciones pueden utilizar una conexión cliente/servidor para acceder a una base de datos Oracle TimesTen, aunque en la mayoría de los casos el mejor desempeño se obtendrá con una aplicación directamente relacionada.

Las operaciones SQL se incorporan en un grupo de bibliotecas compartidas que los desarrolladores vinculan con sus aplicaciones

La tecnología IMDB se implementa como base de datos *in-memory* a la cual acceden aplicaciones, programas utilitarios y procesos del sistema por medio de rutinas de bibliotecas compartidas. Los archivos de disco para los registros y puntos de control (copias de backup) se mantienen opcionalmente para fines de recuperación

Estructuras de Datos que Residen en la Memoria

Las bases de datos *in-memory* son mantenidas en segmentos de memoria compartida dentro del sistema operativo, y contienen todos los datos de usuarios, índices, catálogos del sistema, buffers de registros, bloqueos de tablas y espacio temporario. Múltiples aplicaciones pueden compartir una base de datos TimesTen *in-memory,* y una sola aplicación puede acceder a múltiples bases de datos TimesTen en el mismo sistema.

Figura 2. Bases de datos *In-memory*

Procesos del Sistema

Los procesos de segundo plano brindan servicios para las funciones de inicio, cierre y detección de fallas de aplicaciones en el nivel del sistema, y las funciones de carga, puntos de control y manejo de bloqueos totales en el nivel de base de datos. Hay un solo *TimesTen daemon* (un sistema puede tener múltiples instancias) para todas las instancias y un *subdaemon* para cada base de datos.

Programas Administrativos

Los programas utilitarios son explícitamente invocados por los usuarios, los *scripts* o las aplicaciones para desempeñar servicios como SQL interactivo, copias masivas, backup/restauración, migración de base de datos y monitoreo del sistema.

Puntos de Control y Archivos Testigo

Los cambios en los registros de transacciones y base de datos se escriben en el disco periódicamente. Si una base de datos debe recuperarse, Oracle TimesTen fusiona el *punto de control* de la base de datos en el disco con las transacciones finalizadas que aún están en los archivos testigo. Los sistemas normales de archivos de disco se utilizan para los puntos de control y los archivos testigo.

TECNOLOGÍA DE REPLICACIÓN DE DATOS

Cuando se desea tener disponibilidad casi continua o distribución de la carga de trabajo, la replicación de datos puede configurarse para enviar actualizaciones entre dos o más servidores. Se configura un servidor principal para enviar las actualizaciones, y un servidor suscriptor, para recibirlas; no obstante, un servidor puede ser tanto maestro como suscriptor para la replicación bidireccional. La detección y resolución de conflictos basados en el tiempo se utiliza para establecer prioridad en el caso extremo de que los mismos datos se actualicen en múltiples lugares al mismo tiempo.

Las bases de datos que residen en la memoria se mantienen en segmentos de memoria compartida en el sistema operativo, y contienen los datos de usuarios, índices, catálogos del sistema, buffers de registros, bloqueos de tablas y espacios temporarios. Múltiples aplicaciones pueden compartir una base de datos TimesTen *in-memory,* y una sola aplicación puede acceder a múltiples bases de datos.

Los cambios en los registros de transacciones y base de datos se escriben en el disco periódicamente

Figura 3. Mayor replicación de datos para lograr una alta disponibilidad

Cuando se configura la replicación, el *agente de replicación* inicia el proceso para cada base de datos. Si múltiples bases de datos en el mismo servidor se configuran para la replicación, habrá un agente de replicación distinto para cada base de datos. Cada agente de replicación tiene la capacidad de enviar las actualizaciones a uno o más servidores suscriptores, y de recibir actualizaciones de uno o más servidores maestro. Cada una de estas conexiones se implementa como un *thread* separado de ejecución dentro del proceso del agente de replicación. Los agentes de replicación se comunican a través de *stream sockets* TCP/IP.

Para lograr un máximo rendimiento, el agente de replicación detecta las actualizaciones de la base de datos al monitorear el registro existente de transacciones, y envía actualizaciones a los suscriptores en grupos, de ser posible. Solo las transacciones validadas pueden replicarse. En el nodo suscriptor el agente de replicación actualiza la base de datos a través de una interfase eficaz de bajo nivel, evitando el gasto del nivel SQL.

TECNOLOGÍA CACHÉ

Cuando Oracle TimesTen es utilizado para colocar en caché partes de una base de datos Oracle en una base de datos *in-memory*, se crea una construcción de base de datos para un *grupo en caché* con el fin de retener los datos en caché, las aplicaciones invocan rutinas complementarias de bibliotecas compartidas, y un agente del sistema, el *agente de Caché,* se inicia y realiza las transferencias de todos los datos de manera asíncrona entre el caché y la base de datos Oracle.

Cuando se desea tener disponibilidad casi continua o distribución de la carga de trabajo, se puede configurar la replicación para enviar actualizaciones entre dos o más servidores. Un servidor puede ser tanto maestro como suscriptor para la replicación bidireccional.

Figura 4. Caché adicional para los datos de Oracle

Un grupo en caché es un conjunto de una o más tablas ordenadas con una jerarquía lógica a través de relaciones de claves primarias/externas. Cada tabla en un grupo de caché se relaciona con una tabla de la base de datos Oracle. Una tabla de un grupo de caché puede contener todos o un subgrupo de las filas y columnas en la tabla Oracle relacionada. Los grupos de caché pueden ser creados y modificados por el *Administrador de Caché* basado en browser o por medio de sentencias SQL. Los grupos de caché soportan las siguientes características:

• Las aplicaciones pueden tanto leer y escribir en las tablas de los grupos en caché
• Los grupos en caché pueden actualizarse (incorporar los datos de la Base de Datos Oracle en un grupo de caché) tanto automáticamente como manualmente
• Los grupos de caché pueden eliminarse (propagar actualizaciones de caché a las tablas de la Base de Datos Oracle) tanto automáticamente como manualmente
• Los cambios realizados a cualquier tabla de la Base de Datos Oracle o al grupo de caché pueden rastrearse automáticamente

Cuando las aplicaciones actualizan las filas en un grupo de caché, las filas correspondientes en la Base de Datos Oracle se actualizan de manera sincronizada como parte de la misma transacción o de manera asíncrona inmediatamente después, dependiendo del tipo de grupo de caché que se construya. La configuración asíncrona produce un mayor rendimiento y menores tiempos de respuesta de aplicaciones.

Si una validación en la Base de Datos Oracle falla por alguna razón, la transacción volverá a su estado anterior y se conservará su consistencia, mientras que una configuración asíncrona solo registrará la falla en la Base de Datos Oracle, y la transacción Oracle TimesTen no se restaurará automáticamente, debido a que ya ha sido validada. No obstante, la ventaja de la configuración asíncrona (además del desempeño) es que continuará funcionando incluso si la conexión a la Base de Datos Oracle se ha perdido, y aplicará las actualizaciones a la base de datos Oracle automáticamente cuando se restablezca la conexión.

Cuando la opción *Cache Connect to Oracle* se utiliza para colocar en caché partes de una base de datos Oracle en una base de datos Oracle TimesTen, se crea una construcción de *grupo en caché* para retener los datos en caché; las aplicaciones invocan rutinas complementarias de bibliotecas compartidas, y un agente del sistema, el *agente de Caché,* realiza las transferencias de datos de manera asíncrona entre el caché TimesTen y la base de datos Oracle.

Un grupo de caché es un conjunto de una o más tablas ordenadas con una jerarquía lógica a través de relaciones clave principales/externas. Cada tabla en un grupo de caché se relaciona con una tabla de la base de datos Oracle.

Los cambios que se originan en la Base de Datos Oracle se actualizan en el caché por medio del agente de caché. El agente rastrea y actualiza periódicamente las actualizaciones a los datos de caché.

TECNOLOGÍA IMDB DETALLADA

La tecnología de base de datos *in-memory* ofrece un desempeño extraordinario al cambiar las suposiciones respecto de dónde residen los datos durante el tiempo de ejecución. Al administrar todos los datos en la memoria y realizar optimizaciones para ese entorno, la tecnología de base de datos *in-memory* puede funcionar de manera más eficiente y así ofrecer drásticas mejoras en el rendimiento y la capacidad de respuesta.

¿Pero cuál es la verdadera razón de este desempeño? ¿Podría una aplicación obtener los mismos resultados solamente colocando todos sus datos RDBMS en la memoria principal?

Gran parte del trabajo realizado en RDBMS se realiza según la creencia de que los datos residen principalmente en el disco. Los algoritmos de optimización, la administración de grupos de *buffer* y las técnicas de recuperación indexada son evaluados de acuerdo con este fundamental supuesto.

Oracle TimesTen, por otra parte, *sabe* que los datos residen en la memoria principal y, por consiguiente, puede tomar caminos más directos hacia los datos, reduciendo la longitud de la ruta de código y simplificando tanto el algoritmo como la estructura.

Al comparar estas arquitecturas, existen varias diferencias que claramente ilustran cómo se duplican las mejoras de desempeño. Para graficarlo, se pueden encontrar solo algunas de estas diferencias en los algoritmos de optimización, la administración de grupos de *buffer*, y las estructuras de indexación.

Optimización de Consultas

Los algoritmos de optimización de consultas son distintos para los sistemas basados en memoria que para los sistemas basados en disco. Las decisiones de optimización RDBMS se basan en la idea de que los datos residen ante todo en el disco. En un entorno de tiempo de ejecución dinámico, en donde los datos podrían estar en el disco o en caché en la memoria principal en un momento determinado, los sistemas basados en disco deben suponer el peor escenario posible. Debido a que el I/O de disco es mucho más costoso que el acceso a la memoria, los RDBMS basados en disco deben suponer que los datos residen en el disco. Están optimizados para reducir los cuellos de botella de desempeño, los cuales se producen en el I/O de disco. Si esa es la suposición adoptada, entonces el optimizador basado en disco no siempre creará el plan óptimo para los datos que residen principalmente—o completamente—en la memoria principal.

La tecnología IMDB, por otra parte, es conciente de que los datos residen dentro de la memoria principal y optimiza sus consultas conforme a suposiciones más simples. No es necesario crear el peor escenario sobre la base de la residencia en disco, y por consiguiente, sus estimaciones de costo pueden ser más simples y consistentemente precisas.

Al administrar todos los datos en memoria y optimizar el entorno, la tecnología de base de datos *in-memory* puede funcionar con mucha más eficacia, y así ofrecer drásticas mejoras de desempeño. Debido a que el I/O de disco es mucho más costoso que el acceso a la memoria, los RDBMS basados en disco deben suponer que los datos residen en el disco.

Administración de Grupos de *Buffer*

En arquitecturas RDBMS convencionales, se deben mantener grupos de *buffer* para los datos que han sido colocados en caché en la memoria principal. Cuando el procesador de consultas SQL requiere una página de datos, el método de acceso primero debe buscar el grupo de *buffer* para esos datos en la memoria. Incluso si esos datos se encuentran en el grupo de *buffer*, en muchos casos, deben copiarse

fuera del grupo para su posterior procesamiento. El mantenimiento y la administración de este grupo de *buffer*, junto con copias de datos adicionales, incrementan significativamente la carga que implica hacer que los datos estén disponibles para la aplicación. A pesar de ser necesarios en las soluciones de administración de datos basadas en disco, los grupos de *buffer* son innecesarios en la administración de datos basada en la memoria. Oracle TimesTen no tiene grupos de *buffer*. Los datos ya residen dentro de la memoria principal de modo que el grupo de *buffer* no es necesario. Por consiguiente, se reduce la longitud de la ruta de código y el footprint de motor, se evitan las copias, se simplifican los algoritmos, y los datos se entregan a la aplicación de manera más rápida.

Asimismo, ninguno de estos productos RDBMS está diseñado para revertir dinámicamente estas suposiciones basadas en disco durante el tiempo de ejecución. Las suposiciones basadas en disco se entrelazan muy estrechamente dentro de la base de códigos del sistema para redimensionarlas con algunas sentencias condicionales (*if-then-else)*. Como ejemplo de esta interrelación, las suposiciones basadas en disco pueden verse en las estructuras indexaciones de árboles de las dos arquitecturas.

Estructuras de Índices

Árbol B+

En la típica página de índice de árbol B+ de RDBMS, tanto el valor clave como el puntero a los datos se guardan dentro de una entrada de árbol B+. Un nodo de árbol B+ (página en el disco) es compone de muchas entradas de árbol B+, cada entrada mantiene el valor de los datos de índice y el número de página del próximo nodo de índice adecuado, o el número de página del bloque de disco que mantiene la fila de datos buscada. Esta estructura da como resultado un árbol muy amplio y fijo—que es ideal para la reducción de I/O de disco. En una estructura de árbol B+, el principal objetivo es reducir la cantidad de I/O de disco necesaria para lograr la consulta indexada de los archivos de datos. Un árbol B+ logra esto al (1) mantener los valores clave dentro del mismo nodo de árbol B+ (reduciendo el I/O de disco), y (2) manteniendo tantas entradas de índices dentro del nodo como sean posibles (aumentando la cantidad de entradas de árbol B+ que pueden utilizarse con de un solo I/O).

Esta estructura, a pesar de ser ideal cuando los archivos de índices y datos residen en el disco, no es tan ideal cuando residen en la memoria. Una vez que los datos se mantienen dentro de la memoria, el objetivo de un esquema de indexación debería ser reducir los ciclos del CPU, no de I/O. Los ciclos del CPU se utilizan para expandir y comparar valores de índices comprimidos dentro de un árbol B+. También se utilizan importantes ciclos del CPU para administrar *buffers* que mantienen datos e índices ya leídos en la memoria, desde el disco.

Árboles T

En Oracle TimesTen, el panorama es más simple. Un árbol T se optimiza para el acceso a la memoria principal. Presenta una entrada de índice más económica - tanto en términos de tamaño como de algoritmos - que su pariente, el árbol B+. Las conexiones entre los nodos de árbol T se logran a través de punteros *less-than-or-equal-to* (menor o igual que) y *greater-than* (mayor que). Estos punteros hacen referencia a las ubicaciones de memoria, no a las páginas en el disco. Con solo dos comparaciones, el algoritmo de búsqueda de árbol T sabe si el valor que está buscando se encuentra en el nodo actual o en algún otro lugar de la memoria. Y con cada nuevo direccionamiento indirecto del puntero del nodo de índices, su área de búsqueda se divide por la mitad –definición de una verdadera búsqueda binaria.

A pesar de ser necesarios en las soluciones de administración de datos basadas en disco, los grupos de *buffer* son innecesarios en la administración de datos basada en la memoria. Los datos ya residen dentro de la memoria. Una vez que los datos se mantienen dentro de la memoria, el objetivo de un esquema de indexación debería ser reducir los ciclos del CPU, no de I/O.

Figura 5. Índices de árbol T para las bases de datos que residen en la memoria

En la administración de datos de la memoria principal, el objetivo es reducir los requerimientos de espacio, eliminar el I/O de disco, y reducir los algoritmos, las rutas de código, y el *footprint*. Los árboles T reducen los requerimientos de espacio no solo al mantener el valor clave dentro del mismo código de índice, sino simplemente al apuntar al valor que ya reside en la memoria dentro de la fila de datos. Los árboles T no tienen I/O de disco porque las estructuras de árbol de índices residen todas en la memoria.

Los árboles T reducen la complejidad de código y la extensión de la ruta de código al ser mucho más simples en términos de algoritmos. Con los árboles T, usted obtiene todos los beneficios del acceso basado en árbol B+, pero con importantes ahorros de tamaño y movimiento.

¿Por qué la Diferencia?

Los anteriores ejemplos de diferencias de arquitectura muestran que cuando la suposición de residencia en disco (o la ambigüedad de la ubicación de los datos) se elimina, la complejidad se ve drásticamente reducida. La cantidad de instrucciones de las máquinas se redujo, por lo menos, en un factor de diez, la administración de grupos de *buffer* desapareció, las copias de datos adicionales no son necesarias, se eliminaron las páginas de índices, y se simplificó su estructura. Cuando la residencia en memoria para los datos es la suposición central, todo es más fácil, más elegante, más compacto y más rápido.

Una Nueva Mirada hacia el Rendimiento de Precios

Para muchas aplicaciones, aprovechar la tecnología de base de datos *in-memory* brinda una nueva oportunidad de obtener una ventaja competitiva, mejorando la satisfacción del usuario e impulsando el retorno de la inversión.

Los índices de árbol T se optimizan para las arquitecturas de datos que residen en memoria. Debido a que todas las filas ya se encuentran en la memoria, los árboles T no incluyen valores claves – solamente punteros hacia las filas actuales (los valores clave se muestran en este ejemplo solo para fines explicativos). Cuando se elimina la suposición de residencia en disco, todo es más fácil, más elegante, más compacto y más rápido

Las mejoras de desempeño pueden medirse en múltiplos, en lugar de en fracciones. Una aplicación que utiliza la mitad de su tiempo de procesamiento administrando datos y la mitad de su tiempo dentro del espacio de aplicación o lógica de negocios puede alcanzar casi el doble de capacidad a través del uso de un producto de administración de datos en la memoria. Esta duplicación de capacidad del sistema permite tener alternativas mucho más amplias en términos de estrategias de mercado, ahorros de arquitectura y elección del sistema.

DESEMPEÑO EXCEPCIONAL

Escalabilidad

La tecnología de Oracle TimesTen se construye para aprovechar múltiples CPU en computadoras simétricas de múltiples procesadores. El gráfico de abajo muestra el rendimiento de transacciones de Oracle TimesTen 7.0 en un sistema Linux/Intel de 4 CPUs. Cada transacción ejecuta una sola operación SQL de actualización o selección (lectura), de acuerdo con lo indicado. Los resultados se muestran para 1, 2 y 4 CPUs.

Figura 6. Rendimiento de Oracle TimesTen

La carga de trabajo más simple y, por consiguiente, de mayor volumen es la prueba "100% lecturas", que mide cuántos registros individuales pueden recuperarse utilizando una consulta de valor clave. En este caso, se completaron 253.511 lecturas por segundo, en promedio. Con solo 1 CPU, el índice de lectura fue de 102.707 por segundo.

Las operaciones de actualización requieren más procesamiento que una operación de lectura, en parte porque deben registrarse para fines de recuperación. En este resultado, más de 67.000 registros fueron actualizados por segundo en 4 CPUs y más de 34.000 en 1 CPU.

A pesar de que estas cargas de trabajo puras no son representativas de ninguna aplicación particular de negocios, estos resultados validan la eficiencia extrema de Oracle TimesTen, y cualquier combinación realista de las operaciones SQL posiblemente resulte en un máximo desempeño en el rango "miles a diez miles" por segundo. Si se requieren volúmenes más grandes, TimesTen asciende a un sistema de 4 CPUs. Tenga en cuenta que los resultados de desempeño varían en distintas plataformas de procesadores.

Una aplicación que utiliza la mitad de su tiempo de procesamiento administrando datos y la mitad de su tiempo dentro del espacio de aplicación o lógica de negocios puede alcanzar casi el doble de capacidad.

Oracle TimesTen es capaz de tener un rendimiento excepcional, de diez mil transacciones de actualización a cien mil operaciones de lectura por segundo, cuando la transacción contiene una sola operación SQL. Estos resultados fueron registrados en un sistema Linux/Intel de 4 CPUs con un procesador 3.0 GHz

Tiempo de Respuesta

El rendimiento es un producto derivado de los tiempos de respuesta de las transacciones individuales. A medida que aumenta el rendimiento, el tiempo de respuesta promedio desciende. El rendimiento excepcional de Oracle TimesTen produce tiempos de respuesta de microsegundos en los sistemas actuales a través de una variedad de cargas de trabajo para la administración de datos. Un microsegundo es 1 millonésima parte de un segundo. El tiempo de respuesta promedio para las cargas de trabajo de rendimiento anteriormente descriptas se muestra en el segundo párrafo de abajo.

Figura 7. Latencia de Oracle TimesTen

Es importante tener en cuenta que el tiempo de respuesta de una transacción es el tiempo de respuesta total de todos los pasos que constituyen el procesamiento, como la administración de datos, I/O, la conexión en red y la lógica de negocios. El desempeño de una parte de Oracle TimesTen no es el único determinante del tiempo de respuesta total. Algunas veces uno de los pasos, generalmente I/O o la conexión en red, dominará al resto.

Por ejemplo, si se implementa una conexión cliente/servidor en red en lugar de una conexión directa entre las aplicaciones y el administrador de datos, se pueden agregar unos milisegundos o más tiempo de respuesta en cada recorrido a través de la red. De igual manera, si se utiliza I/O sincronizado para registrar cambios, la lentitud relativa de una operación de disco será incluida en el tiempo de respuesta. En ambos casos, si el componente de administración de datos se completa en cuestión de microsegundos, el tiempo total de respuesta de la aplicación aún será un factor de la red y las latencias I/O.

Para las aplicaciones que tienen un componente de administración de datos dominante, la aceleración de la tecnología de Oracle TimesTen podría lograr una importante reducción del tiempo de respuesta. La aplicación de mayor desempeño será la que esté directamente conectada, sea capaz de utilizar un registro asíncrono y se base en la administración de datos. Los sistemas de reserva online basados en las búsquedas, los servicios de comunicación basados en las localizaciones y los motores de ruteo de los centros de contacto son buenos ejemplos.

Como siempre, los resultados de desempeño variarán enormemente según distintos factores, como el tamaño de la transacción, la lógica de acceso a elementos no basados en datos y el desempeño de la plataforma.

La latencia de Oracle TimesTen (tiempo de respuesta) se mide en microsegundos (la millonésima parte) para operaciones de un solo registro. Las aplicaciones basadas en datos son las que más se beneficiarán, ya que el tiempo de respuesta total para la aplicación incluirá mucho más que solo la parte de administración de datos. Estos resultados fueron registrados en un sistema Linux/Intel de 4 CPUs con un procesador de 3.0 GHz

FUNCIONALIDAD EN TIEMPO REAL

Esta sección ofrece una visión funcional de los productos de Oracle TimesTen, y describe las interfases y características utilizadas por los desarrolladores al escribir aplicaciones en tiempo real.

Administración de Datos

Modelo de Base de Datos Relacional

Como en el modelo relacional, los datos de los usuarios se almacenan en filas (registros) dentro de tablas. Una tabla se define como un listado de columnas (campos de datos), cada uno con un tipo de datos específico. Los índices se crean explícita o implícitamente para acelerar las búsquedas de rangos de valores o basadas en claves en las tablas. Oracle TimesTen soporta los índices *hash* y, como se describió

anteriormente, los índices de Árbol T. Las restricciones definidas, como la unicidad o la integridad de referencia (relaciones principales-secundarias entre las tablas) se imponen automáticamente.

Dentro de una tabla, se puede designar una clave primaria, compuesta por una o más columnas. Una clave primaria es el identificador exclusivo de una fila, y Oracle TimesTen prohibirá automáticamente la inserción de filas adicionales con el mismo valor de clave primaria. También puede designarse una o más claves externas por tabla, cada una relacionada con la clave primaria de otra tabla. Las claves externas imponen una relación principal-secundaria entre dos tablas. Por ejemplo, una fila con una clave primaria no se eliminará si una clave externa relacionada en otra tabla tiene el mismo valor.

Las vistas materializadas son estructuras opcionales que pueden definirse. Una vista materializada es una tabla de solo lectura que proviene de tablas regulares (básicas). Puede incluir una parte o la totalidad de los datos de una o más tablas, y puede incluir valores calculados (como totales y promedios). Debido a que las vistas materializadas se actualizan automáticamente por el sistema cuando es necesario un cambio en una tabla básica, ellas brindan a las aplicaciones acceso más rápido a los datos frecuentemente obtenidos. Sin las vistas materializadas, se incurrirá en gastos de uniones y otros cálculos cada vez que una aplicación deba leer los datos. Las vistas materializadas son especialmente útiles para soportar el procesamiento de eventos (vea los próximos comentarios).

Si se utiliza *Cache Connect to Oracle*, se pueden definir uno o más "grupos de caché" en una base de datos *in-memory*. Un grupo de caché es un grupo jerárquico de tablas (impuestas a través de claves externas) que se mapea al grupo correspondiente de tablas de Oracle Database, y contiene parte o la totalidad de los mismos datos. Los grupos en caché se leen y escriben, y las actualizaciones a los datos pueden propagarse automáticamente entre Oracle TimesTen y Oracle Database.

Las operaciones de administración de datos se expresan por medio del estándar *Structured Query Language* (SQL). Las operaciones más importantes son SELECT (leer), UPDATE, INSERT, y DELETE. También se soporta la operación de unión relacional, que combina columnas de más de una tabla con un grupo de resultados, como la mayoría de las opciones SQL definidas en la definición estándar SQL92 de base de línea.

Las tablas, los índices, las vistas materializadas y los grupos en caché son creados y mantenidos dentro de las bases de datos. Las aplicaciones se conectan a una base de datos TimesTen y requieren operaciones de administración de datos frente a los datos que estén dentro ella. Pueden existir múltiples bases de datos en el mismo sistema informático, y se puede acceder a ellas por la misma aplicación. No obstante, ninguna operación exclusiva (por ej., una unión entre dos tablas) puede abarcar más de una sola base de datos. Las aplicaciones pueden participar en una transacción distribuida, a través del soporte de la interfase XA estándar (o JTA para Java) de Oracle TimesTen para validaciones de dos fases.

El administrador de datos de Oracle TimesTen implementa un verdadero modelo de base de datos relacional. No es un modelo híbrido o un modelo diferente con una vista relacional encima. Los desarrolladores que conocen los populares productos RDBMS deberían ser inmediatamente productivos.

Un grupo de caché es una grupo jerárquico de tablas que se mapea hacia el grupo correspondiente de tablas de Oracle Database, y contiene parte o la totalidad de los mismos datos. Los grupos en caché se leen y escriben, y las actualizaciones a los datos pueden propagarse automáticamente entre Oracle TimesTen y Oracle Database.

El debate anterior de la tecnología IMDB de Oracle TimesTen explicaba cómo los servicios de administración de datos se brindan en "tiempo real". Vale la pena repetir que no es que toda la base de datos reside en la memoria y así elimina I/O –en realidad no tiene que ver con eso. El hecho de que se *garantice* que los datos residen en la memoria permite la eliminación de las instrucciones de procesamiento que, de lo contrario, deberían justificar la ambigüedad de la localización de los datos. Esto incluye algoritmos de búsqueda indirecta, gastos de administración de *buffer*, copias de datos, y estructuras de índices subopcionales y planes de ejecución.

Asimismo, la configuración de la información dentro de una base de datos Oracle TimesTen se optimiza para obtener rápido procesamiento y conservación de memoria. En RDBMS basado en disco, las estructuras de disco y de memoria son casi idénticas; una página de memoria es del mismo tamaño que

un tamaño de bloque I/O – para minimizar las demoras de I/O. Con TimesTen, eso no es importante, de modo que las estructuras de memoria pueden designarse para minimizar las instrucciones del procesamiento.

Durabilidad y Concurrencia

Oracle TimesTen se ajusta a las propiedades "ACID" de un sistema de administración de datos: atomicidad, consistencia, aislamiento y durabilidad. Estas propiedades garantizan que, en un sistema de múltiples usuarios, cada transacción funciona como si fuera la única transacción que se ejecuta en ese momento, y que el sistema puede garantizar que los efectos de una transacción validada no se perderán. Estas son las propiedades más estrictas requeridas para un administrador de datos, y TimesTen tiene la capacidad de conformidad total.

Una mala percepción muy común es que los administradores de datos en memoria no pueden evitar la pérdida de datos debido a las fallas del sistema. De hecho, *las mismas técnicas que hacen que las transacciones y los datos sean duraderos en una base de datos convencional se utilizan en TimesTen*. Como en todos los sistemas orientados a las transacciones, la durabilidad se logra a través de la combinación de los registros de cambio y las actualizaciones periódicas de una versión de base de datos que reside en disco.

La principal diferencia es que Oracle TimesTen ofrece control de aplicaciones sobre el intercambio entre el grado de durabilidad y el rendimiento total. Tener más de un elemento implica tener menos de otro, y Oracle TimesTen ofrece alternativas para este espectro.

Operaciones de Disco

Si una aplicación requiere que los cambios no se pierdan, los registros de archivo deben eliminarse del disco como parte de la validación de la transacción. Si el máximo desempeño es más importante que la posibilidad de perder algunas transacciones, los registros de archivo pueden escribirse en el disco con menos frecuencia, de manera asíncrona desde cada validación de transacción. En cualquier caso, el administrador de datos de Oracle TimesTen intentará "agrupar validaciones" de múltiples transacciones para reducir las escrituras de disco.

Dentro de la base de datos, se crean y mantienen tablas, índices, vistas materializadas y grupos de caché. Puede haber múltiples bases de datos en el mismo sistema informático, y se puede acceder a ellas por la misma aplicación. Oracle TimesTen se ajusta a las propiedades "ACID" de un sistema de administración de datos: atomicidad, consistencia, aislamiento y durabilidad.

Una mala percepción muy común es que los administradores de datos en memoria no pueden evitar la pérdida de datos debido a las fallas del sistema. De hecho, las mismas técnicas que hacen que las transacciones y los datos sean duraderos en una base de datos convencional se utilizan en TimesTen. La principal diferencia es que TimesTen ofrece control de aplicaciones sobre el intercambio entre el grado de durabilidad y el rendimiento total.

En forma periódica, la base de datos se controla automáticamente en el disco sobre la base del volumen de registro y la frecuencia especificada por el usuario. TimesTen soporta operaciones "confusas" de puntos de control que funcionan en el segundo plano con mínimo impacto en la ejecución de las aplicaciones. La recuperación de una falla del sistema implica fusionar los registros de archivo con el archivo de punto de control más reciente.

Las bases de datos en Oracle TimesTen pueden crearse de manera permanente o temporaria, y pueden designarse para acceso exclusivo (es decir, un solo usuario) o compartido (múltiples usuarios). Las bases de datos permanentes (archivos de punto de control) permanecen en el disco cuando no se utilizan. Las bases de datos temporarias residen en la memoria, y son eliminadas cuando no se utilizan.

El registro de transacciones cumple con dos propósitos. Primero, el registro permite la recuperación de transacciones frente a las bases de datos persistentes después de que ocurre una falla del sistema. Segundo, el registro permite al administrador de datos Oracle TimesTen realizar una detección y eliminación completa cuando se utiliza en modo de acceso compartido. Una base de datos no registrada ofrece un mejor desempeño. No obstante, con los registros desactivados, las transacciones no pueden recuperarse y volver a su estado anterior y los bloqueos solo pueden realizarse en el nivel de base de datos (no se permite el bloqueo en el nivel de fila o de tabla). Desactivar el registro solo es adecuado

para las operaciones de un usuario único que pueden reiniciarse desde el principio en caso de que ocurra alguna falla.

Las bases de datos permanentes, compartidas, registradas en disco son las configuraciones más comúnmente implementadas. Las bases de datos temporarias son comunes en las aplicaciones para las cuales los datos son tan transitorios que no tiene sentido recuperar la base de datos después de una falla del sistema. La información dinámica del "estado" en una aplicación de un centro de llamadas es un ejemplo.

Bloqueo y Aislamiento

Oracle TimesTen aplica los bloqueos para evitar que un usuario cambie los datos actualmente leídos o cambiados por otro usuario. Un bloqueo es una "reserva" en los objetos de la base de datos ya sea en la base de datos, las tablas o el nivel de filas.

Los bloqueos del nivel de filas ofrecen la mayor concurrencia de usuarios múltiples, y se aplican por defecto para las bases de datos de Oracle TimesTen. Las aplicaciones pueden invocar un procedimiento para cambiar el nivel de bloqueo de la fila o base de datos durante el tiempo de ejecución. El bloqueo en el nivel de tabla se utiliza cuando el optimizador TimesTen determina qué es ventajoso, o si la aplicación invoca un procedimiento que lleva al optimizador a aplicar el bloqueo del nivel de tabla durante la duración de una transacción.

El "aislamiento" especifica el comportamiento de los bloqueos que se están aplicando en respuesta a las operaciones de lectura. Las aplicaciones pueden elegir entre dos niveles de aislamiento: en serie o de lectura validada (por defecto).

Las bases de datos en Oracle TimesTen pueden crearse de manera permanente o temporaria y pueden designarse para el acceso exclusivo o compartido.

Se puede aplicar un bloqueo en los objetos de base de datos, ya sea en el nivel de filas, tablas o base de datos. Los bloqueos en el nivel de filas ofrecen la mayor concurrencia de usuarios múltiples, y Oracle TimesTen lo aplica por defecto

El aislamiento en serie se utiliza para las transacciones que exigen valores de datos consistentes y predecibles durante el ciclo de vida de una transacción. Con el asilamiento en serie, los registros que se leen no pueden ser actualizados ni eliminados por otros usuarios hasta que la transacción se valide o vuelva a su estado anterior. También durante este tiempo, el usuario no puede insertar nuevos registros si ellos han sido parte del resultado de estas operaciones de lectura. En otras palabras, el aislamiento en serie garantiza que una operación de lectura podría repetirse con los mismos resultados. Con el bloqueo del nivel de base de datos, las transacciones funcionan efectivamente con el aislamiento en serie por defecto.

El aislamiento de lectura validada produce la mayor concurrencia de usuarios múltiples. Garantiza que los lectores solo verán los valores de datos validados, no espera a que los responsables de escrituras liberen los bloqueos, ni establece bloqueos en los registros que se están leyendo. Para lograr esto, Oracle TimesTen crea dos copias de un registro que está siendo actualizado: los valores preactualizados y "validados" para los lectores y la versión actualizable para los responsables de la escritura. Los lectores no tienen el acceso bloqueado a la versión de "lectura", y los responsables de escritura no están pendientes de los lectores.

Procesamiento de Consultas

La mayoría de los productos especializados diseñados para lograr un alto desempeño requieren el uso de APIs de propiedad, "protegidas". Oracle TimesTen, no. El objetivo primario siempre ha sido la adopción de estándares abiertos relevantes. SQL (lenguaje de consulta estructurado), JDBC (conectividad de base de datos Java) ODBC (conectividad de base de datos abierta) y JMS (Servicio de Mensajería Java) son los estándares principales, y la única manera de acceder a la base de datos TimesTen. La implementación de estas interfaces ha sido altamente adaptada a la arquitectura de Oracle TimesTen.

Durante años, SQL ha sido ampliamente adoptado como el estándar de lenguajes de consulta para bases de datos relacionales. La abstracción de los detalles subyacentes de almacenamiento e indexación es uno de los principales beneficios de SQL. Es también un lenguaje fácil de utilizar que expresa "qué" hacer, no "cómo". SQL no tiene referencia a los índices, tipos de datos ni la disposición física– solo tablas, columnas y condiciones de búsqueda. La característica de optimización de Oracle TimesTen determina la manera más rápida de responder a la consulta basándose en factores como la presencia de índices y la distribución de valores clave.

Este nivel de abstracción permite que el modelo de datos subyacente se ajuste o extienda sin afectar las aplicaciones existentes. Los nuevos servicios pueden incorporarse rápidamente en el entorno de producción simplemente al agregar módulos de aplicaciones y cualquier tabla o columna de datos adicional. Sin un lenguaje de consulta como SQL que proteja la aplicación del administrador de datos, la mayoría de las aplicaciones serían afectadas por la incorporación de nuevos campos de información.

Las aplicaciones pueden elegir entre dos niveles de aislamiento: en serie o de lectura validada

El aislamiento de lectura validada genera la mayor concurrencia de usuarios múltiples. Garantiza que los lectores solo vean los valores de datos validados, pero no espera que los escritores eliminen los bloqueos ni bloqueen los registros que se están leyendo

El objetivo primario de TimesTen es adoptar estándares abiertos relevantes. SQL, JDBC, ODBC y JMS son las interfaces a través de las cuales las aplicaciones interactúan con las bases de datos Oracle TimesTen. La implementación de estas interfaces ha sido altamente adaptada a la arquitectura TimesTen

Replicación de Datos

La replicación es el proceso de copiar información entre las bases de datos. El motivo fundamental detrás de la replicación de Oracle TimesTen es hacer que los datos estén continuamente disponibles para las aplicaciones críticas con un mínimo impacto en el rendimiento. Además de su rol en la recuperación ante fallas, la replicación también es útil para distribuir las cargas de usuarios a través de múltiples bases de datos, a fin de lograr un máximo rendimento y facilitar la actualización y el mantenimiento online.

Oracle TimesTen respalda un modelo de replicación "maestro-suscriptor", en el cual los cambios validados se copian desde la fuente a una o más bases de datos suscriptoras. La replicación se configura mediante sentencias SQL y puede aplicarse a tablas designadas o a toda una base de datos. Para permitir una alta eficiencia y pocos gastos generales, se utiliza un esquema de replicación basado en registros de transacción.

La replicación en cada base de datos maestro o suscriptora es controlada por agentes de replicación que se comunican mediante *stream sockets* TCP/IP. El agente de replicación en la base de datos maestro lee los registros desde el registro de transacciones y reenvía cualquier cambio detectado en los elementos replicados al agente de replicación en la base de datos suscriptora. El agente de replicación suscriptor luego aplica las actualizaciones a su base de datos local. Si el agente suscriptor no se está ejecutando cuando las actualizaciones son reenviadas por la secuencia maestra, ésta retiene las actualizaciones en su registro de transacción hasta que se apliquen en el suscriptor.

Equilibrio de Desempeño y Consistencia

Por defecto, la replicación Oracle TimesTen es un mecanismo asíncrono. Al utilizar la replicación asíncrona, una aplicación actualiza una base de datos maestro y continúa funcionando sin esperar que las actualizaciones sean recibidas por el suscriptor. Las bases de datos maestro y suscriptoras tienen mecanismos internos para confirmar que las actualizaciones hayan sido recibidas con éxito y hayan sido validadas por el suscriptor. Estos mecanismos garantizan que las actualizaciones sean aplicadas en un suscriptor solo una vez, pero son totalmente invisibles para la aplicación.

La replicación asíncrona brinda un desempeño máximo, pero la aplicación es completamente desconectada del proceso de recepción de los elementos replicados en el suscriptor. Oracle TimesTen también ofrece dos opciones en el *servicio de recepción* para las aplicaciones más "pesimistas" que necesitan mayores niveles de confianza respecto de que los datos replicados sean consistentes entre las bases de datos maestro y suscriptoras:

• El *servicio de aviso de recepción* de acopla débilmente o "sincroniza" la aplicación con el mecanismo de replicación al bloquear la aplicación hasta que la replicación confirma que la actualización ha sido recibida por el suscriptor
• El *servicio de recepción twosafe* permite la replicación totalmente sincronizada al bloquear la aplicación hasta que la replicación confirme que la actualización ha sido recibida *y validada* en el suscriptor

La replicación se configura mediante sentencias SQL y puede aplicarse a tablas designadas o a toda una base de datos. Para permitir una alta eficiencia y pocos gastos generales, TimesTen utiliza un esquema de replicación basado en un registro de transacción

Las bases de datos maestro y suscriptoras tienen mecanismos internos para confirmar que las actualizaciones han sido recibidas con éxito y validadas por el suscriptor. Estos mecanismos garantizan que las actualizaciones sean aplicadas en el suscriptor solo una vez, pero son totalmente invisibles para las aplicaciones

Las aplicaciones que utilizan los servicios de recepción intercambian el rendimiento para garantizar altos niveles de integridad de datos y consistencia entre las bases de datos maestro y suscriptoras. En caso de una falla maestro, la aplicación tiene un grado más alto de confianza respecto de que una transacción validada en la base de datos maestro persista en la base de datos de suscripción. La replicación confirmación de recepción tiene un menor impacto de desempeño que *twosafe* a expensas de una menor sincronización.

Topologías de Replicación

Al designar una base de datos como maestro y como suscriptora, puede configurarse la replicación bidireccional. Además, Oracle TimesTen permite la replicación multimodo de "N direcciones", ofreciendo una amplia gama de topologías de replicación posibles, cubriendo tanto las configuraciones *hot-standby* como activa-activa. Esta última también puede desglosarse en una carga de trabajo dividida (donde cada tabla replicada tiene un solo maestro) y una carga de trabajo distribuida (donde una tabla tiene más de un maestro). En una configuración de carga de trabajo distribuida, la aplicación tiene la responsabilidad de dividir el trabajo entre los sistemas, de manera que no se produzcan choques de replicación. En caso de que se produzcan choques, un mecanismo de detección y resolución de choques basado en *time-stamp* evita las replicas inconsistentes.

Figura 8. Opciones para la configuración de replicación

Usted también puede establecer que un suscriptor sirva de *propagador* para recibir las actualizaciones replicadas de un maestro y pasarlas a sus propios suscriptores. Los propagadores son útiles para optimizar el desempeño de replicación sobre conexiones de red de menor banda ancha, como aquellas entre los servidores y una Intranet.

La replicación asíncrona ofrece un desempeño máximo. TimesTen también brinda dos opciones para el "servicio de recepción" para las aplicaciones que necesitan mayores niveles de confianza respecto de que los datos replicados sean consistentes entre las bases de datos maestro y suscriptoras

Caché

Integración personalizada

Es común para Oracle TimesTen implementarse junto con un RDBMS basado en disco, por medio del cual TimesTen es utilizado cuando los datos necesitan capturarse o procesarse en tiempo real. Como los datos transitan a un estado que no es en tiempo real (por ej., cuando se finaliza la negociación de stock o se clasifica el registro detallado de una llamada) la información es transferida de TimesTen al RDBMS de backend. Hay muchas formas por las cuales puede lograrse esta integración.

Las aplicaciones pueden conectarse a Oracle TimesTen y a la base de datos de backend y realizar el movimiento de datos mediante solicitudes API regulares. Este es el enfoque más flexible, pero es el menos transparente para las aplicaciones, y podría requerir una programación compleja. Por ejemplo, una aplicación que necesita poner en caché todos los suscriptores activos podría primero solicitar un registro en Oracle TimesTen, y si el resultado es "no encontrado", conectarse al RDBMS de backend y repetir la misma solicitud, luego ingresar los resultados en Oracle TimesTen. El acceso posterior a esa

información es muy rápido. Si hubiese alguna actualización para esos datos, debería repetirse en ambas bases de datos. La dificultad recae en los cambios realizados directamente a los datos en el RDBMS de backend, dando como resultado problemas de coherencia en caché.

Un derivado del enfoque de arriba es conectar Oracle TimesTen y el RDBMS de backend mediante una vía de interconexión de mensajes para publicar y suscribir, y escribir nuevos módulos separados de las aplicaciones existentes que "escuchen" los cambios a publicar en la vía de interconexión y dupliquen los cambios recogidos de esa vía de interconexión. Una aplicación puede utilizar una API para registros de transacción (XLA) de Oracle TimesTen para registrar y recibir avisos de actualizaciones realizadas a la base de datos. La mayoría de los RDBMS ofrecen una característica *trigger* que puede utilizarse para señalar cambios, o suministrar una API al registro de transacción, similar a XLA.

Cache Connect to Oracle

Una opción más fácil y transparente para integrar las bases de datos Oracle TimesTen y las bases de datos Oracle es mediante la incorporación de Cache Connect to Oracle, con sus conexiones incorporadas.

A diferencia de la mayoría de los cachés, que son de solo lectura, Cache Connect to Oracle respalda el caché de lectura/escritura para los datos Oracle y la propagación bidireccional de actualizaciones entre Oracle TimesTen y Oracle Databases. Otra distinción es el concepto de un "grupo caché", que describe un grupo de tablas Oracle TimesTen que se equiparan a todas o a un subgrupo de tablas en Oracle Database. Un grupo caché puede consistir en todas o un subgrupo de filas y columnas en estas tablas de Oracle Database.

Cache Connect to Oracle brinda controles para especificar cuánto tiempo deberían permanecer los datos de Oracle Database en caché. Además de la capacidad para configurar funciones automáticas, un grupo caché puede cargarse, actualizarse, purgarse y descargarse a pedido mediante sentencias SQL.

"Los proveedores líderes de servicio confían en nuestros productos para administrar los datos de facturación críticos, por lo tanto, no se admiten cuellos de botella ni tiempos de baja. Con Oracle TimesTen y Oracle Real Application Clusters, obtenemos una base confiable, escalable, probada y en tiempo real para la administración de datos a fin de brindar un mejor servicio a nuestros clientes"

—Ed McKee,
Director de Aplicaciones, Interact, Inc.

A diferencia de la mayoría de los cachés, que son de solo lectura, Cache Connect to Oracle respalda el caché de lectura/escritura de datos Oracle y la propagación bidireccional de actualizaciones mediante Oracle TimesTen y Oracle Database

Cache Connect to Oracle interactúa con la base de datos Oracle para realizar todas las operaciones sincronizadas del grupo caché, como crear un grupo caché, cargar el grupo caché desde Oracle, y propagar las actualizaciones entre el grupo caché y Oracle. Asimismo, existe un proceso de segundo plano, denominado *Oracle Agen*t, que realiza todas las operaciones caché asíncronas, como propagar automáticamente las actualizaciones de Oracle Database a un grupo caché en Oracle TimesTen.

Las aplicaciones Cache Connect to Oracle pueden enviar las sentencias SQL a un grupo caché o a Oracle Database mediante una sola conexión. Esta capacidad de conexión única es activada por una característica *pass-through* que controla si la sentencia SQL puede ser administrada localmente por las tablas caché o si debe ser redireccionada a Oracle Database. Los datos en caché pueden actualizarse en el grupo caché de Oracle TimesTen o en Oracle Database. Cache Connect to Oracle ofrece la capacidad de propagar automáticamente las actualizaciones del grupo caché a Oracle Database, así como de Oracle Database al grupo caché en TimesTen.

Figura 9. Característica de conexión única de Cache Connect to Oracle

Hay dos categorías básicas de los grupos caché:

• Los grupos caché **administrados por sistema** ofrecen comportamientos de caché predeterminados que son administrados enteramente por Cache Connect to Oracle.

• Los grupos caché **administrados por usuarios** permiten que los usuarios elijan entre todos los atributos y sentencias SQL para definir comportamientos caché personalizados. Los usuarios tienen control total sobre los mecanismos de carga, antigüedad y propagación.

Hay dos tipos básicos de grupos caché administrados por sistema:

• Los grupos caché **writethrough** cargan los datos de tabla en caché desde Oracle Database una vez creados. A partir de entonces, todas las actualizaciones al grupo caché se propagan automáticamente a Oracle. Los grupos caché *writethrough* pueden ser asíncronos (AWT) o sincrónicos (SWT). Los grupos caché SWT esperan la validación en Oracle Database antes de realizar la validación en el caché TimesTen. Los grupos caché AWT validan los cambios en el caché sin esperar una validación en Oracle Database.

Las aplicaciones pueden enviar sentencias SQL a un grupo caché o a Oracle mediante una sola conexión. Esta capacidad de conexión única es activada por una característica *pass-through* que verifica si la sentencia SQL puede ser administrada localmente por las tablas en caché o si debe redireccionarse a Oracle

• Los grupos caché **readonly** contienen datos de solo lectura, de manera que las actualizaciones en las tablas del grupo caché no están permitidas. Las actualizaciones deben realizarse en Oracle Database. Estas actualizaciones pueden realizarse directamente en la base de datos Oracle o mediante TimesTen al redireccionar las actualizaciones a la base de datos Oracle utilizando la característica *passthrough*. Por defecto, un grupo caché READONLY es automáticamente actualizado cuando se producen las actualizaciones en la base de datos Oracle. La frecuencia de actualización es controlada por la aplicación.

Figura 10. Grupos caché administrados por sistema y por usuarios

Procesamiento de Eventos

El procesamiento de eventos es tener la capacidad de "detectar y responder" a eventos considerados relevantes para los negocios.

De acuerdo con una investigación reciente de Gartner[1] sobre la empresa en tiempo real (RTE), "La RTE monitorea, captura y analiza la causa de origen y los eventos explícitos que son críticos para el éxito en el momento que ocurren esos eventos, para identificar nuevas oportunidades, evitar cualquier contratiempo y minimizar las demoras en los procesos importantes de negocio. La RTE luego aprovechará esa información para eliminar progresivamente las demoras en la administración y ejecución de sus procesos importantes de negocio".

Hay varias formas mediante las cuales los productos Oracle TimesTen activan las aplicaciones para el procesamiento de eventos. En el nivel más básico, son diseñadas para respaldar aplicaciones que residen al margen de la red, donde se generan los mensajes y eventos de negocio y donde pueden detectarse primero. La capacidad de residir junto con las aplicaciones en una amplia gama de servidores informáticos brinda la flexibilidad de respaldar aplicaciones para el procesamiento de eventos.

---

[1] Gartner Actualiza su Definición de Empresa en Tiempo Real, K. McGee, 25 de marzo de 2004

El comportamiento de grupos caché puede predefinirse y estar administrado por sistemas gracias a Cache Connect to Oracle, o controlado por usuarios para lograr una flexibilidad máxima. Los grupos caché administrados por sistema implementan configuraciones predefinidas de solo lectura o *writethrough*

No obstante, a menudo los eventos simples no muestran su relevancia, y deben correlacionarse con otra información u organizarse en etapas hasta que un patrón de eventos sea evidente. En estos casos, los

datos del evento deben capturarse en un administrador de datos, lo cual puede posiblemente compararse con los datos de referencia que se extrajeron de otra fuente de datos. Cuando se identifica un evento que requiere medidas, tener la respuesta adecuada debe, en muchos casos, ser instantáneo. Dicho procesamiento es idealmente adecuado para la funcionalidad de los productos Oracle TimesTen.

Oracle TimesTen Transaction Log API (XLA) permite la detección de las actualizaciones realizadas a una base de datos. Las aplicaciones utilizan XLA para monitorear los cambios en una base de datos TimesTen y tomar medidas de acuerdo con esos cambios. Por ejemplo, se puede necesitar notificación cuando:

• El valor de negociación total de cualquier cliente ha excedido los $1.000.000 en el día
• Un cliente de la lista "A" realiza una compra
• Se agota un saldo prepago
• Un elemento de red está offline
• Un vuelo cambia de puerta de embarque

Múltiples aplicaciones pueden leer simultáneamente las actualizaciones del registro de transacciones, y cada aplicación puede mantener su propia "señal" en el registro de actividades para mantener su posición. Las señales son persistentes a través de las conexiones de base de datos, las desconexiones y las fallas del sistema, de manera que las notificaciones de eventos pueden capturarse donde se dejaron. XLA es a menudo utilizado para crear una solución personalizada para la replicación de datos en una base de datos que no sea de Oracle TimesTen.

La funcionalidad para el procesamiento de eventos de Oracle TimesTen puede lograrse en un RDBMS convencional a través del uso de *triggers* y procedimientos almacenados. Sin embargo, XLA está diseñado para tener menos gastos generales y un mejor desempeño, consistente con las expectativas en tiempo real. También, cuando se combina con la funcionalidad de vistas materializadas, los eventos XLA pueden apuntar a subgrupos muy granulares de la base de datos. Los *triggers* tradicionales de base de datos ejecutan su lógica cada vez que se produce un cambio en algún registro de la tabla.

Las vistas materializadas, junto con la funcionalidad XLA, ofrecen una manera eficiente de implementar la notificación detallada de eventos. Es decir, una vista materializada recopila, en un solo lugar, los datos relacionados con eventos específicos sobre los cuales se pueden tomar medidas. Una aplicación XLA solo necesita monitorear los registros de actualización interesantes de una sola tabla con vistas materializadas. Sin una vista materializada, la aplicación XLA debería monitorear todos los registros de actualización de todas las tablas base, incluidos los registros que reflejan actualizaciones en filas y columnas poco interesantes para la aplicación.

CONCLUSIÓN

Con la gran velocidad de los mensajes que pasan a través de las redes empresariales, el uso del procesamiento en tiempo real para capturar, analizar y responder inteligentemente a eventos clave se está convirtiendo, cada vez más, en el estándar de comparación para la excelencia corporativa. Esto no solo es importante para la ejecución y la administración de procesos de negocio críticos. Los clientes esperan interacciones altamente adaptadas y la máxima capacidad de respuesta de cualquier empresa con la cual realizan negocios significativos.

Lo que se necesita es una generación de software de infraestructura liviana que tenga interfaces conocidas y poderosas, y lenguajes de consulta ampliamente utilizados– que puedan interconectarse fácilmente con las bases de datos, los sistemas de mensajería y los servidores de aplicación existentes– y explote todo el potencial de desempeño de las actuales plataformas informáticas en red, que tienen mucha memoria. Esta es la generación de software de infraestructura para la administración de datos en tiempo real ofrecida por Oracle TimesTen.

Los productos de Oracle TimesTen brindan administración de datos en el nivel de aplicaciones para sistemas con desempeño crítico, optimizados para lograr respuestas rápidas y caché en tiempo real de los datos Oracle. Cientos de compañías alrededor del mundo utilizan Oracle TimesTen en aplicaciones de producción, con inclusión de Amdocs, Aspect, Avaya, la Bolsa de Valores de Bombay, Cisco, Ericsson, JP Morgan, Lucent, NEC, Nokia, Salesforce.com y Sprint.

Productos y Tecnologías de Oracle TimesTen
Febrero de 2007

Oracle Corporation
Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Consultas:
Teléfono: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2007, Oracle. Todos los derechos reservados.
El presente documento tiene solo fines informativos y su contenido está sujeto a cambios sin que medie notificación alguna. El presente documento puede contener errores y no está sujeto a ninguna otra garantía ni condición, ya sea oral o que se encuentre implícita en la ley, con inclusión de garantías y condiciones implícitas de comerciabilidad o aptitud para un fin específico. En especial, negamos cualquier responsabilidad con respecto al presente documento, el cual no crea obligación contractual alguna, sea en forma directa o indirecta. El presente documento no podrá ser reproducido ni transmitido de ninguna forma ni por ningún medio, sea electrónico o mecánico, con ningún fin, sin que hayamos otorgado previamente nuestro consentimiento por escrito. Oracle, JD Edwards, PeopleSoft y Siebel son marcas registradas de Oracle Corporation y/o sus afiliadas. Otros nombres pueden ser marcas comerciales de sus respectivos propietarios.