

Documento técnico de Oracle
Julio de 2009

Uso de Oracle In-Memory Database Cache para acelerar la base de datos de Oracle

1. Introducción.....	2
2. Almacenamiento en caché de la capa de aplicaciones	4
3. Base de datos Oracle TimesTen In-Memory.....	6
3.1 Rendimiento de Oracle TimesTen	8
4. Almacenamiento de datos en caché con Oracle In-Memory Database Cache	9
4.1 Definición del contenido de una caché	10
4.2 Carga de datos y administración de la memoria caché	12
4.3 Datos compartidos en una malla de caché	14
4.4 Mantenimiento de la coherencia de los datos	14
4.5 Alta disponibilidad	20
5. Rendimiento	24
6. Ejemplos	26
6.1 Caché de sólo lectura	26
6.2 Caché de sólo lectura con intervalo móvil.....	27
6.3 Caché actualizable.....	27
6.4 Caché dinámica actualizable	28
6.5 Caché de captura de datos con tasa de ingreso irregular	29
6.6 Caché de captura de datos con tasa de ingreso constantemente alta.....	29
6.7 Caché actualizable gestionada por el usuario	30
6.8 Caché de sólo lectura, dinámica y distribuida	31
7. Conclusión	31
8. Referencias	32

1. Introducción

Oracle In-Memory Database Cache acelera los procesos comerciales, permite la inteligencia de negocios en tiempo real y facilita la personalización de las aplicaciones en las tiene lugar el diálogo con el cliente.

El producto *Oracle In-Memory Database Cache (IMDB Cache)* es una opción ideal de Oracle Database para el almacenamiento en caché de subparticiones esenciales para el rendimiento de una base de datos de Oracle en la capa de aplicaciones. El uso de IMDB Cache mejora el tiempo de respuesta y la capacidad de procesamiento de las aplicaciones. IMDB Cache cuenta con tres componentes tecnológicos clave: la tecnología de Oracle TimesTen In-Memory Database (TimesTen) para administrar los datos en tiempo real en la capa de aplicaciones; la tecnología de almacenamiento en caché para almacenar las tablas de acceso frecuente desde un servidor de Oracle Database a la capa de aplicaciones y mantener la coherencia de los datos almacenados; y un componente de replicación de datos transaccionales para asegurar una alta disponibilidad entre capas.

TimesTen es una base de datos relacionales de memoria optimizada que ofrece un tiempo de respuesta muy breve y un procesamiento muy alto de los sistemas de rendimiento crítico. Está diseñada para operar en la capa de aplicaciones, cerca de las aplicaciones y, en forma opcional, en proceso con aplicaciones. Una base de datos TimesTen puede utilizarse como registro o como caché de una base de datos de Oracle.

Las aplicaciones pueden crear y administrar tablas de bases de datos en TimesTen o almacenar en caché las subparticiones de una base de datos de Oracle en IMDB Cache. Las tablas almacenadas en caché y las que no lo están pueden coexistir en la misma base de datos de memoria, y además son todas persistentes y recuperables. Las consultas y actualizaciones de datos almacenados en caché y no almacenados se ejecutan con aplicaciones mediante SQL92 o PL/SQL que usan ODBC, JDBC, la interfaz Oracle Call Interface (OCI) o TTClasses, así como también Pro*C.

Para la escalabilidad horizontal del rendimiento y la capacidad, es posible valerse de mallas de caché. En ese caso, la malla consiste en una recopilación de cachés de IMDB que administran en forma conjunta los datos almacenados en caché de una aplicación.

Los datos almacenados en caché se distribuyen entre los miembros de la malla, que brinda aplicaciones con transparencia de ubicación, lo que causa que los datos almacenados en caché se agreguen eficazmente en todos los miembros de la malla disponibles para la aplicación. Las mallas de caché permiten la escalabilidad incremental mediante la adición (y eliminación) en línea de los miembros de la malla. Mantienen la coherencia de los datos copiados en caché entre los miembros de la malla de caché y la base de datos de Oracle.

IMDB Cache administra la disponibilidad de los datos a través de la capa de aplicaciones y la capa del servidor de la base de datos. Asegura una alta disponibilidad y no causa pérdidas de transacciones, independientemente de que se produzca una falla, ya sea un error en uno de los nodos de caché, uno de los nodos de Oracle RAC, al nivel de la red o incluso en el caso de un grupo RAC.

TimesTen y IMDB Cache cuentan con antecedentes comprobados de implementaciones de producción en empresas en tiempo real e industrias en las que el tiempo es crítico, que incluyen servicios de telecomunicaciones de redes, sistemas de soporte operacional, centros de contacto, sistemas de reserva y líneas aéreas, sistemas de control y comando, y comercio de valores. Miles de compañías en todo el mundo usan TimesTen e IMDB Cache en aplicaciones de producción, incluidos Alcatel-Lucent, Amdocs, Aspect, Avaya, Bombay Stock Exchange, Bridgewater Systems, BroadSoft, Cisco, Deutsche Börse, Ericsson, JP Morgan, NEC, NYFIX, Smart Communications y Sprint.

2. Almacenamiento en caché de la capa de aplicaciones

El almacenamiento en caché de la capa de aplicaciones suele utilizarse con el fin de mejorar la latencia del acceso a los datos y disminuir la carga de trabajo en la base de datos interna.

Se desarrollaron diversas técnicas de este tipo de almacenamiento para mejorar el rendimiento del acceso a los datos o disminuir la conflictividad de los servidores de la base de datos interna. Un tiempo de respuesta rápido es especialmente importante en las aplicaciones en tiempo real y en las que tiene lugar el diálogo con el cliente. La disminución de la carga de trabajo en la base de datos interna también es importante para las aplicaciones que cuentan con una comunidad de usuarios en crecimiento, tales como los servicios de software en hosts, sitios de comercio por medios electrónicos o servicios de telecomunicación.

Existen diversas opciones con respecto a qué información almacenar en caché y dónde almacenarla, y cada opción tiene sus ventajas y desventajas. Algunas de las técnicas de almacenamiento en caché creadas son:

- *Cachés de los resultados de las consultas.* Generalmente se realiza en la capa de aplicaciones y se administra con un software especial que oculta la presencia de la caché a los ojos de la aplicación. En esta situación, el software para almacenamiento en caché guarda automáticamente los resultados de las consultas enviadas al sistema de la base de datos. Una coincidencia de caché se reconoce y ejecuta desde la caché si una consulta es idéntica a otra enviada anteriormente, incluidos iguales valores de parámetros. Las ventajas del almacenamiento en caché son su sencillez y que ofrece escenarios de acceso en los que la misma consulta puede enviarse varias veces. Sin embargo, su alcance es limitado ya que no puede manejar el proceso de las consultas en el contenido del caché.
- *Caches de herramientas de asignación de objetos relacionales.* Las herramientas de asignación de objetos relacionales (herramientas de asignación de O/R) ocultan las bases de datos relacionales a los ojos de los programadores orientados a objetos, mediante una asignación entre los objetos y los datos relacionales. Una vez asignados los datos relacionales a la representación de un objeto, pueden almacenarse con la herramienta de asignación de O/R hasta que no se necesiten más o hasta que sean obsoletos. El almacenamiento en caché mediante las herramientas de asignación de O/R es una técnica habitual usada para evitar costosas asignaciones entre el modelo del objeto del lenguaje de programación y el modelo relacional de la base de datos.
- *Cachés de objetos.* La frase "almacenamiento en caché" no es muy apropiada en este caso, ya que los objetos que se guardan en estas cachés no son necesariamente particiones de objetos almacenados en otros lugares. Estas "cachés" son repositorios de objetos independientes del origen de los objetos. Por lo general, no son transparentes para las aplicaciones. Las

aplicaciones "ponen", "toman", "insertan" y "eliminan" objetos en las cachés. No hay muchos programas en el mercado que ofrezcan estos tipos de cachés, y el nivel de funcionalidad que admiten no es siempre el mismo. Las cachés pueden encontrarse estrictamente en la memoria o pueden copiarse en un disco o en otro sistema de administración de datos. Algunos productos brindan control de concurrencia, otros ofrecen distribución transparente en los diversos nodos de una red y otros cuentan con alta disponibilidad.

Oracle In-Memory Database Cache posee funcionalidad SQL y relacional completa, mantenimiento automático de la coherencia de los datos con la base de datos de Oracle y rendimiento en tiempo real.

Además, Oracle In-Memory Database Cache (IMDB Cache) proporciona un enfoque único, ya que permite el almacenamiento de tablas o fragmentos de tablas en caché desde una base de datos de Oracle a la capa de aplicaciones. Los fragmentos de la tabla se describen mediante una sintaxis SQL extendida y se almacenan en caché en la base de datos Oracle TimesTen In-Memory Database (TimesTen). Las aplicaciones leen y actualizan los datos almacenados con SQL, PL/SQL o Pro*C, e IMDB Cache propaga automáticamente las actualizaciones desde la base de datos de Oracle a la memoria caché y viceversa.

Una serie de IMDB Caches puede configurarse como una malla de caché. Los datos almacenados en caché se distribuyen entre los miembros de la malla, y la malla de caché brinda aplicaciones con transparencia de ubicación y control de concurrencia, lo que hace que todos los datos almacenados en caché se agreguen eficazmente en los miembros disponibles para las aplicaciones. Cuando es necesario que la capacidad o el rendimiento de una aplicación aumente, pueden agregarse nodos complementarios a la malla de caché sin interrumpir el servicio. De este modo, IMDB Cache brinda a las aplicaciones la funcionalidad y generalidad completas de una base de datos relacional, escalabilidad incremental junto con transparencia de ubicación, mantenimiento automático de la coherencia de la memoria caché con la base de datos de Oracle y el rendimiento en tiempo real de una base de datos de memoria.

El enfoque de IMDB Cache cuenta con dos beneficios importantes que favorecen el rendimiento general. Primero, las aplicaciones que usan la opción IMDB Cache experimentan una reducción significativa en el tiempo de respuesta y un aumento en el procesamiento a causa de la arquitectura de memoria de TimesTen y la eliminación de la comunicación entre la capa de aplicaciones y el servidor de la base de datos. Después, este enfoque reduce la carga de trabajo en la base de datos interna, lo cual mejora el procesamiento general de todas las aplicaciones.

La capacidad de brindar todas las ventajas de las bases de datos relacionales junto con el rendimiento en tiempo real, la escalabilidad incremental y la administración automática de la memoria caché son características únicas que sólo ofrece IMDB Cache. Es ideal para el almacenamiento en caché de las subparticiones de una base de datos de Oracle, fundamentales en

el rendimiento, lo que permite tanto lecturas como actualizaciones de los datos almacenados en caché y la administración automática de la coherencia de los datos.

En las próximas secciones se presentará una breve introducción a Oracle TimesTen In-Memory Database (puede encontrarse información más detallada en [1]), una descripción de cómo Oracle In-Memory Database Cache almacena datos en caché y los administra, y algunos escenarios de almacenamiento en caché ilustrativos.

3. Base de datos Oracle TimesTen In-Memory

La base de datos TimesTen In-Memory brinda acceso transaccional a los datos y funcionalidad relacional mediante interfaces API estándar.

Oracle TimesTen In-Memory Database es una base de datos relacional de memoria optimizada que admite SQL92 y PL/SQL mediante ODBC, JDBC, Oracle Call Interface (OCI) e interfaces TTClasses¹, así como también Pro*C/C++. Como admite interfaces estándar y difundidas de Oracle, TimesTen asegura una fácil adopción de las aplicaciones existentes.

Aunque TimesTen funciona con datos que se encuentran en la memoria principal, las bases de datos TimesTen son coherentes y recuperables en caso de que falle el suministro eléctrico, el software o el hardware. La durabilidad se asegura mediante los puntos de control y el registro en disco. Las aplicaciones pueden elegir propiedades ACID en sus transacciones, pero también pueden optar por alternativas más flexibles para un rendimiento mayor. TimesTen ofrece un optimizador de consultas basado en el costo y las aplicaciones pueden visualizar e influir en los planes de las consultas. La base de datos TimesTen está disponible como una biblioteca que puede vincularse mediante aplicaciones, así como también a través de una opción de cliente/servidor. Cuando se accede a TimesTen mediante la opción de cliente/servidor, cada solicitud a TimesTen incurre en gastos por la comunicación entre procesos aun si la aplicación y el servidor de TimesTen se ejecutan en la misma máquina. Por el contrario, cuando TimesTen se vincula con la aplicación, las solicitudes a TimesTen son sólo llamadas locales que implican un gasto general insignificante, y las transferencias de datos entre la aplicación y TimesTen son operaciones de copiado de memoria poco costosas. La alta disponibilidad se obtiene mediante la replicación. También se ofrecen diversas utilidades, tales como una SQL interactiva, una herramienta gráfica para el desarrollo de la base de datos y la configuración de caché, restauración

¹ TimesTen C++ Interface Classes (TTClasses) es una biblioteca de clase C++ que brinda envolturas alrededor de la funcionalidad más común ODBC. Es más sencillo de utilizar que ODBC y fomenta las mejores prácticas al mismo tiempo que mantiene un rendimiento veloz.

y backup en línea, y carga en bloque. Las operaciones de mantenimiento de la base de datos también pueden encontrarse con interfaces API programáticas.

Al momento de la ejecución, se guarda una copia de la base de datos en la memoria principal. Se administra en un segmento de la memoria compartido al que acceden todos los procesos conectados a esa base de datos. En la figura 1, puede observarse la arquitectura de un sistema de base de datos de memoria TimesTen.

Los algoritmos y las estructuras de datos de Oracle TimesTen In-Memory Database se optimizan alrededor de la residencia de la memoria de los datos.

Los algoritmos de acceso y las estructuras de datos de TimesTen explotan la residencia de la memoria de la base de datos a fin de obtener un gran avance en el rendimiento. En comparación con una base de datos basada en disco totalmente almacenada en caché, la arquitectura de memoria optimizada de TimesTen utiliza muchos menos ciclos de CPU porque se elimina la sobrecarga derivada de la administración de búferes de memoria y de responder por las ubicaciones de múltiples datos (disco y memoria).

El rendimiento de memoria optimizada de Oracle TimesTen se complementa con la funcionalidad que admite propiedades transaccionales, mecanismos de persistencia y recuperación de fallas del sistema. Una variedad de opciones se encuentra disponible para el bloqueo, el aislamiento y registro de los diversos usuarios, lo que permite una gama de escenarios de aplicaciones desde cachés de búsqueda momentáneas a sistemas de comercio financieros y de facturación de telecomunicaciones.

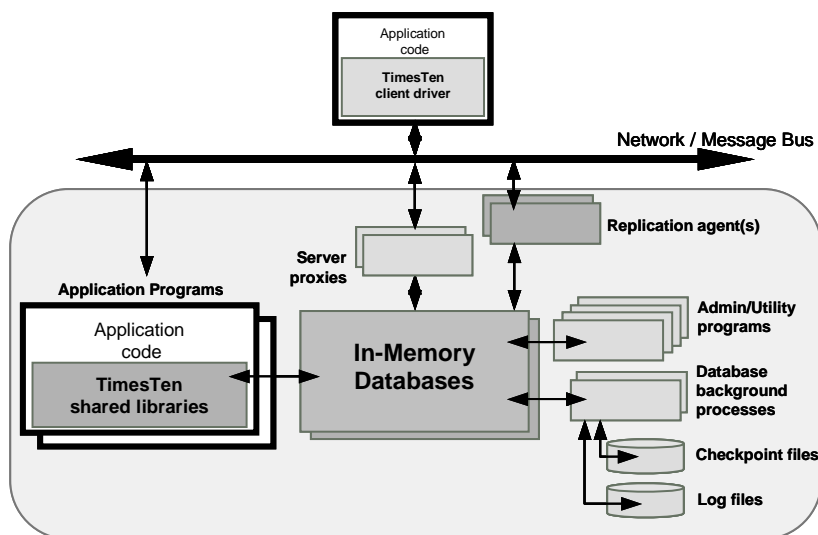


Figura 1. Arquitectura de TimesTen

Las bases de datos TimesTen son persistentes y recuperables.

En TimesTen, la durabilidad se obtiene por el registro de los cambios de transacciones ejecutadas en el disco y la actualización periódica de una imagen del disco de la base de datos con puntos de control. El momento de escritura del registro en el disco puede configurarse con la aplicación, ya sea en sincronía con el fin de la transacción o en forma diferida hasta un tiempo después, lo que brinda un mejor rendimiento. En muchas situaciones, es preferible un mayor procesamiento que el registro sincrónico, especialmente cuando el valor monetario de una transacción es bajo o los datos tienen una vida útil corta, por ejemplo, cuando se rastrea la ubicación de teléfonos celulares en una red que comunica la ubicación del celular en pocos segundos.

TimesTen permite a las aplicaciones realizar un seguimiento de los cambios en tablas específicas. Es muy útil en entornos donde las aplicaciones son susceptibles a ciertos eventos. Por ejemplo, una aplicación que quiere saber si el precio de cierta acción superó determinado límite. Esta función de notificación de cambios es especialmente útil, ya que permite el seguimiento de los cambios producidos en las tablas de las bases y en las visualizaciones materializadas también.

3.1 Rendimiento de Oracle TimesTen

Los tiempos de respuesta muy breves no pueden lograrse mediante adiciones de hardware. TimesTen ofrece una latencia muy breve gracias a su arquitectura exclusiva.

Además, puede alcanzar tiempos de respuesta en microsegundos con su arquitectura de memoria. Con TimesTen, una transacción que lee el registro de una base de datos puede tardar menos de 5 microsegundos y las transacciones que actualizan o insertan un registro pueden realizarse en menos de 15 microsegundos.

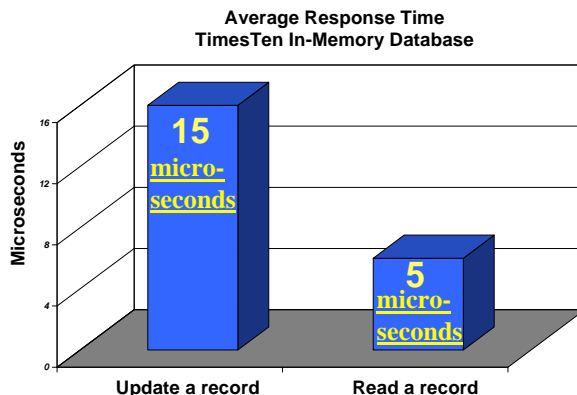


Figura 2. Tiempo de respuesta de TimesTen

En la figura 2 se muestran los tiempos de respuesta de una aplicación que ejecuta transacciones de lectura y actualización en un sistema 2-CPU Intel E5450 (8 vías/3 GHz) que ejecuta Oracle Enterprise Linux 5.2.

4. Almacenamiento de datos en caché con Oracle In-Memory Database Cache

IMDB Cache contiene subparticiones de las tablas de una base de datos de Oracle.

Permite el almacenamiento en caché de las tablas de subparticiones de una base de datos de Oracle a la capa de aplicaciones. Las tablas almacenadas en caché pueden actualizarse e IMDB Cache sincroniza los datos entre la base de datos de Oracle y la memoria caché.

El motor de base de datos que administra los datos almacenados en caché es Oracle TimesTen In-Memory Database. Es mejor gracias a la capacidad de carga y sincronización de los datos almacenados en caché. Uno de los procesos internos asociados con IMDB Cache es el agente de caché, que administra parte de esta sincronización. En la figura 3 puede observarse la arquitectura de una memoria IMDB Cache.

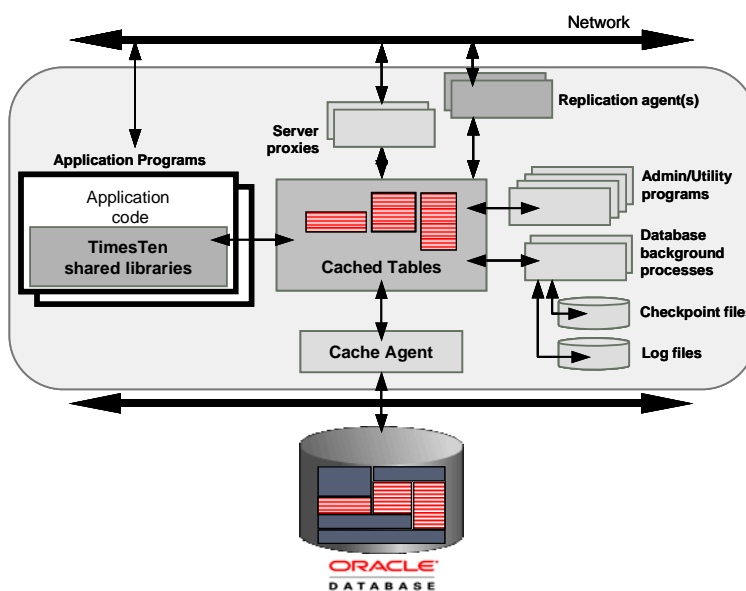


Figura 3. Arquitectura de IMDB Cache

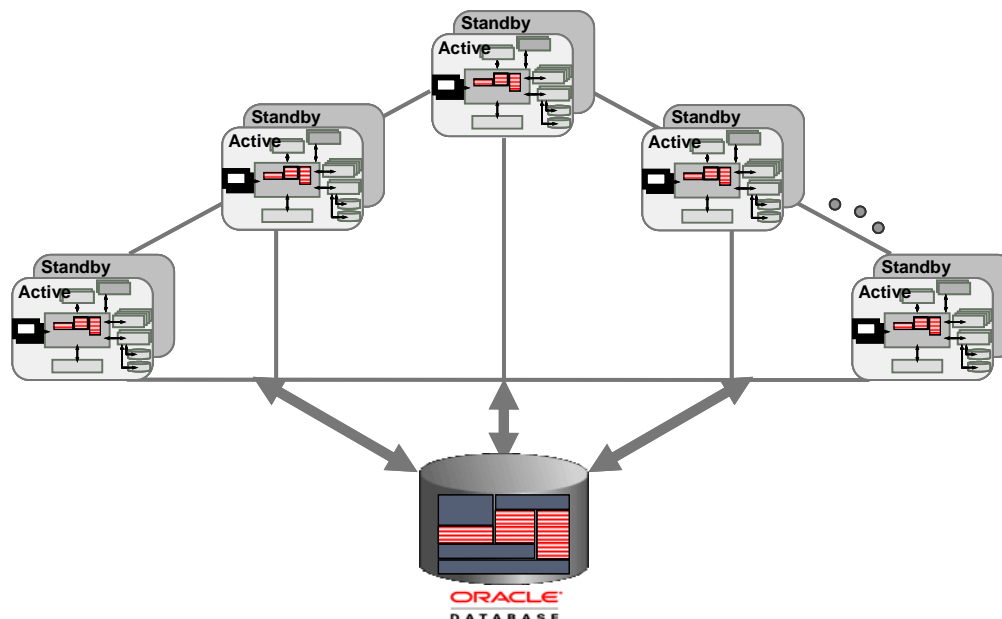


Figura 4. Malla de caché con cinco miembros de malla replicados

La *malla de caché* es un conjunto de IMDB Cache que administra en forma colectiva la información de las aplicaciones. Está compuesta por uno o más miembros de malla, cada uno de los cuales está respaldado por una instancia IMDB Cache. Los miembros de malla almacenan en caché las tablas de una base de datos de Oracle central o de Real Application Cluster (RAC). Los datos almacenados en caché se distribuyen en múltiples nodos o IMDB Caches sin compartir el almacenamiento. La malla de caché garantiza que la información sea coherente en todos los nodos. Los miembros de malla se pueden replicar.

La figura 4 muestra una malla de caché compuesta por cinco miembros de malla replicados. Se pueden agregar miembros de malla adicionales sin interrumpir las operaciones. La configuración de replicación que se debe utilizar con IMDB Cache es la configuración de pares activos de reserva.

4.1 Definición del contenido de una caché

El contenido de Oracle IMDB Cache se define mediante la sintaxis extendida de SQL.

El *grupo de caché* es un conjunto de tablas de IMDB Cache que corresponde a un conjunto de tablas de bases de datos de Oracle utilizadas frecuentemente, las cuales se relacionan a través de restricciones de claves externas. La sintaxis SQL se utiliza para definir grupos de caché y elegir las columnas y las filas que se deberán almacenar en caché a partir de las tablas de la base de datos

de Oracle. Los usuarios pueden definir los grupos de caché en forma programática o a través de la utilidad `ttlsq` interactiva.

Ejemplo:

Supongamos que en la base de datos de Oracle existen las siguientes tablas:

- Cliente (ID de cliente, Nombre, Edad, Sexo, Dirección, Estado, Código Postal, Número de teléfono)
- Pedido (ID de cliente, ID de pedido, Fecha de compra, Importe)
- Interés de cliente (ID de cliente, Interés)

Una aplicación puede querer colocar en caché los perfiles de los clientes que realizaron pedidos desde el 1 de enero de 2009. Para eso, puede definir los dos grupos de caché que se detallan a continuación:

- El primer grupo de caché contiene subgrupos de las tres tablas ya mencionadas arriba sobre clientes que realizaron pedidos desde el 1 de enero de 2009 y que además viven en EE. UU., en la región del Pacífico. Además, la aplicación puede elegir colocar en caché sólo un subgrupo de las columnas de las tablas. Por ejemplo, puede colocar en caché las siguientes columnas:
 - Cliente (ID de cliente, Nombre, Edad, Sexo, Estado)
 - Pedido (ID de cliente, ID de pedido, Fecha de compra, Importe)
 - Interés de cliente (ID de cliente, Interés)
- El segundo grupo de caché contiene la misma información que el primer grupo de caché, pero acerca de los clientes de la región montañosa de Estados Unidos.

Los dos grupos de caché se pueden colocar en caché, en diferentes nodos que ejecutan IMDB Cache.

Otro concepto que utiliza IMDB Cache es el de "instancia caché". *Instancia caché* es un conjunto de registros relacionados y de identificación única, y se utiliza para modelar un objeto complejo. Las instancias caché forman la unidad de carga y antigüedad de la memoria caché, tal como se describe más abajo. En el ejemplo anterior, todos los registros de las tablas Cliente, Pedido e Interés de cliente que pertenecen a una ID de cliente determinada, pertenecen a la misma instancia caché y se relacionan entre sí a través de restricciones de claves externas. La ID de cliente identifica la instancia caché en forma única y se denomina *clave de instancia caché*.

TimesTen es compatible con los mismos tipos de datos que Oracle Database.

Además de admitir sus propios tipos de datos, TimesTen admite los mismos tipos de datos básicos que la base de datos de Oracle, por lo cual no es necesario relacionar los tipos de datos

de Oracle Database con los tipos de datos de TimesTen. Sin embargo, es posible relacionar los tipos de datos de Oracle Database para que la implementación de TimesTen tenga una mayor eficiencia. Por ejemplo, una aplicación puede relacionar un tipo de datos NUMÉRICO de Oracle Database a un tipo de datos ENTERO de TimesTen.

Tenga en cuenta que los desarrolladores de aplicaciones pueden crear índices en las tablas de memoria caché. Los índices de memoria caché y los índices de la base de datos de Oracle pueden ser iguales o diferentes. El diseñador de la aplicación puede utilizar la flexibilidad de TimesTen para crear muchos índices en la misma tabla y puede definir índices sobre varias columnas.

4.2 Carga de datos y administración de la memoria caché

Una aplicación debe decidir cómo cargar los datos de grupo de caché en IMDB Cache para su procesamiento. Para la carga de datos, se pueden utilizar las siguientes técnicas:

- Carga explícita. Se puede realizar de distintas formas:
 - Cargar todo el grupo de caché de una sola vez. Esta técnica puede emplearse en caso de que el contenido de todo el grupo de caché entre en la memoria caché. También existe la opción de descargar un grupo de caché completo.
 - Cargar instancias caché “por medio de la cláusula WHERE”. En este caso, se utiliza una cláusula WHERE para describir el subgrupo de las instancias caché que deberían cargarse en la memoria caché. Las aplicaciones también pueden descargar las instancias caché “por medio de la cláusula WHERE”.
 - Cargar instancias caché “por medio de la cláusula ID”. En este caso, se utiliza una lista de ID de instancias caché para especificar las instancias caché que deben cargarse en la memoria caché. Las aplicaciones también pueden descargar las instancias caché “por medio de la cláusula ID”.
- Carga dinámica. Esta técnica se encuentra disponible para la carga de instancias caché. La carga dinámica resulta muy útil cuando el grupo de caché es demasiado grande y no cabe en la memoria caché. Por eso, la memoria caché sólo puede guardar el conjunto de trabajo de la aplicación. En este caso, los registros que componen la instancia caché se cargan automáticamente en la memoria caché por un fallo de la memoria caché, por ejemplo,

cuando una sentencia SQL² no encuentra los datos solicitados en la memoria caché. Si la instancia caché ya se encuentra en la memoria caché, la sentencia se maneja directamente desde la memoria caché.

La carga dinámica normalmente se asocia con la *antigüedad de caché* automática. Las instancias caché pueden determinarse automáticamente como obsoletas y eliminarse cuando se excede la capacidad de la memoria caché. IMDB Cache es compatible con los mecanismos de *antigüedad por uso* y *antigüedad por tiempo*. El mecanismo de antigüedad por uso utiliza un esquema LRU (último utilizado) para identificar las instancias caché que hace más tiempo que no se utilizan y eliminarlas cuando se supere la capacidad de la caché. El mecanismo de antigüedad por tiempo otorga a las instancias de caché una *vida útil* de cierta duración en la memoria caché. Para que este mecanismo funcione, es necesario que haya una columna de marca de fecha y hora en una de las tablas del grupo de caché. La aplicación gestiona el valor de la columna de marca de fecha y hora. Las instancias caché pueden permanecer en la memoria caché mientras que el valor de marca de fecha y hora sumado a la vida útil no sobrepase el momento actual. Observe que la antigüedad de caché se puede utilizar independientemente de la carga dinámica. De hecho, se puede utilizar con tablas normales de TimesTen que no se hayan almacenado en la memoria caché desde la base de datos de Oracle.

Una aplicación puede tener algunos grupos de caché sujetos al mecanismo de antigüedad y otros no. Por ejemplo, la aplicación puede mantener información de catálogos todo el tiempo en la memoria caché y cargar los perfiles de los usuarios según se necesite, cuando los usuarios ingresan a la aplicación. Luego, puede eliminar los perfiles automáticamente, cuando los usuarios se desconectan. La aplicación también puede descargar las instancias caché en forma explícita.

Los datos cargados en las tablas de la memoria caché se encuentran disponibles para el procesamiento de SQL, PL/SQL y Pro*C mediante JDBC, ODBC, TTCclasses y OCI.

² La carga dinámica de una instancia caché se encuentra disponible para las sentencias SQL con una expresión de igualdad en la clave principal o externa de cualquiera de los registros de la instancia caché.

4.3 Datos compartidos en una malla de caché

Los grupos de caché pueden ser locales o globales. En el caso de los *grupos de caché locales*, los datos colocados en caché no se comparten entre los miembros de una misma malla de caché. Los miembros de la malla pueden tener datos inconexos o superpuestos. En este caso, la aplicación determina la distribución de los datos entre ellos. Por ejemplo, los datos de catálogos de sólo lectura pueden almacenarse en la memoria caché de todos los miembros de la malla para un mejor rendimiento y los datos actualizables de clientes pueden dividirse por zonas geográficas entre miembros de distintas mallas. Las actualizaciones ejecutadas en tablas de caché se propagan a las tablas de Oracle sin coordinación con otros miembros de malla. Un grupo de caché local puede definirse como de carga explícita o de carga dinámica. En forma predeterminada, los grupos de caché son locales, a menos que se los defina como globales.

En el caso de un *grupo de caché global*, los datos colocados en caché se comparten entre los miembros de una misma malla de caché. El control de concurrencia se implementa en toda la malla y cualquier transacción que se realice en cualquier parte de la malla siempre se encontrará con la última versión ejecutada de una instancia caché. Las actualizaciones ejecutadas por diferentes miembros de una malla para una misma instancia caché se propagan hacia la base de datos de Oracle en el orden de ejecución dentro de la malla, para asegurar la coherencia de los datos.

4.4 Mantenimiento de la coherencia de los datos

Oracle IMDB Cache es compatible con actualizaciones de datos almacenados en caché y mantiene la coherencia entre las memorias caché y la base de datos de Oracle en forma automática.

Los datos almacenados en caché pueden actualizarse en IMDB Cache o en la base de datos de Oracle. IMDB Cache brinda la opción de propagar automáticamente las actualizaciones desde la memoria caché hacia la base de datos de Oracle y viceversa. Sin embargo, se presupone que el grupo de caché se actualiza siempre o la mayoría de las veces en la memoria caché, o bien en la base de datos de Oracle. Una gran falla de diseño consiste en almacenar en caché un grupo de tablas para que se actualicen frecuentemente tanto en la memoria caché como en la base de datos interna. No obstante, existen casos en los que resulta apropiado realizar actualizaciones en ambas. Por ejemplo, las actualizaciones en la base de datos de Oracle pueden realizarse sólo durante las noches por razones de mantenimiento, mientras que las actualizaciones en las caché se realizan de día. También puede ocurrir que las actualizaciones de los datos centrales se realicen en la base de datos de Oracle y que las actualizaciones de los datos regionales se realicen en las caché.

Los grupos de caché pueden ser *gestionados por el sistema* o *gestionados por el usuario*. Existen tres tipos de grupos de caché gestionados por el sistema:

- *Grupos de caché de sólo lectura.* Estos grupos de caché no pueden actualizarse en la memoria caché. Pueden actualizarse en la base de datos de Oracle. IMDB Cache gestiona la propagación de las actualizaciones desde la base de datos de Oracle hacia la caché.
- *Grupos de caché con introducción asincrónica (AWT).* Estos grupos de caché pueden actualizarse en la memoria caché pero no en la base de datos de Oracle. IMDB Cache propaga las actualizaciones desde la caché hacia la base de datos de Oracle en forma asincrónica luego de ejecutada una transacción.
- *Grupos de caché con introducción sincrónica (SWT).* Estos grupos de caché pueden actualizarse en la memoria caché pero no en la base de datos de Oracle. Las actualizaciones de las tablas de la memoria caché se propagan hacia la base de datos de Oracle en forma sincrónica con la ejecución de una transacción.

Los grupos de caché gestionados por el sistema cuentan con semánticas bien definidas y con restricciones para imponer dichas semánticas. Por el contrario, la semántica de los grupos de caché gestionados por el usuario depende de la aplicación. Por ejemplo, un grupo de caché gestionado por el usuario puede ser actualizable tanto en la memoria caché como en la base de datos de Oracle.

Los grupos de caché de sólo lectura, con introducción asincrónica, con introducción sincrónica y gestionados por el usuario pueden ser todos locales. Sin embargo, sólo los grupos de caché con introducción asincrónica dinámicos se pueden especificar como grupos de caché globales.

En la siguiente tabla se resumen las distintas opciones de carga de grupos de caché, formas de compartir datos entre mallas de caché y opciones para mantener la coherencia que se encuentran disponibles.

		Carga de datos en un grupo de caché			
		Carga explícita		Carga dinámica	
Mantenimiento de la coherencia de los datos	Grupo de caché de sólo lectura	x		x	
	Grupo de caché con introducción asincrónica	x		x	x
	Grupo de caché con introducción sincrónica	x		x	
	Grupo de caché gestionado por el usuario	x		x	
		Grupo de caché local	Grupo de caché global	Grupo de caché local	Grupo de caché global

Datos compartidos en una malla de caché

Las aplicaciones de IMDB Cache pueden enviar sentencias SQL tanto a un grupo de caché como a la base de datos de Oracle mediante una sola conexión con una base de datos IMDB Cache. La opción de esa conexión única es posible gracias a la función *PassThrough*, que verifica si la sentencia SQL se puede gestionar localmente mediante las tablas en la memoria caché o si es necesario redireccionarla hacia la base de datos de Oracle. La función *PassThrough* ofrece configuraciones que especifican el tipo de sentencias que han de derivarse y en qué circunstancias. Una configuración muy útil es la que especifica que todas las sentencias que actualizan la base de datos deben derivarse a la base de datos de Oracle. Esta configuración permite que las actualizaciones de una aplicación se ejecuten en la base de datos de Oracle y las lecturas se realicen en IMDB Cache mediante una sola conexión.

Las siguientes secciones describen las operaciones de IMDB Cache que se encuentran disponibles para mantener la coherencia de los datos almacenados en caché. IMDB Cache inicia automáticamente algunas de esas operaciones, mientras que el inicio de otras está a cargo la aplicación en forma explícita.

4.4.1 Propagación de actualizaciones de IMDB Cache a la base de datos de Oracle y entre miembros de mallas de caché para grupos de caché globales

Como ya vimos, los grupos de caché globales también son grupos de caché con introducción asincrónica y dinámicos. Una aplicación con grupos de caché globales se conectará con uno de los miembros de la malla. Por lo general, accederá a instancias caché que ya han sido almacenadas en caché en el miembro de la malla. Sin embargo, si intentara acceder a una instancia caché que no se encontrara en el miembro de malla, IMDB Cache cargará esa instancia caché dinámicamente tanto desde otro miembro del grupo o desde la base de datos de Oracle, según dónde resida la versión actualizada más reciente de la instancia caché. Esto sucede automáticamente, sin la intervención de la aplicación. IMDB Cache determina dónde reside la copia más reciente y utiliza comunicación entre pares para intercambiar información con otras bases de datos IMDB Cache de la malla.

Si una transacción actualiza una instancia caché en alguno de los miembros de malla, estará disponible el siguiente mecanismo para mantener la base de datos de Oracle en sincronía con el caché:

- *Propagate*. IMDB Cache propaga las actualizaciones a la base de datos de Oracle una vez ejecutada la transacción. Si, al poco tiempo, otra transacción actualiza la misma instancia caché en otro miembro de malla y se ejecuta, IMDB Cache garantiza que las ejecuciones se propaguen a la base de datos de Oracle en el orden correcto.

La figura 5 muestra una malla de caché compuesta por tres miembros de malla. Todos los miembros de malla almacenan en caché los datos del mismo grupo de caché global y cada miembro de malla tiene en su caché únicamente unas pocas instancias caché del grupo de caché global. Estas instancias se almacenan en caché porque se ha accedido a ellas recientemente en sus correspondientes miembros de malla. A medida que pasa el tiempo, el acceso a cada instancia puede seguir a través de sus miembros de malla y, por lo tanto, permanecer allí; seguir en otro miembro de malla y la instancia se traslada a ese miembro; o ser nulo, en cuyo caso se supera totalmente la antigüedad de la malla de caché.

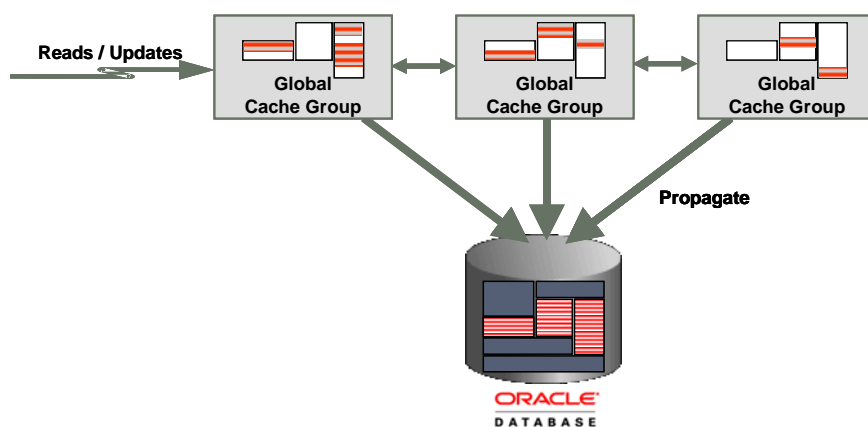


Figura 5. Propagación de actualizaciones y coherencia de caché para grupos de caché globales

4.4.2 Propagación de actualizaciones de IMDB Cache a la base de datos de Oracle para grupos de caché locales

Para los grupos de caché locales que se pueden actualizar en la caché, se encuentran disponibles los siguientes mecanismos a fin de mantener la base de datos de Oracle en sincronía con la caché:

- *Propagate*. Con la opción de propagación activada, todas las modificaciones realizadas a un grupo de caché (por ejemplo, todas las operaciones de inserción, actualización y eliminación) se propagan automáticamente a la base de datos de Oracle. El momento en el que la propagación se lleva a cabo difiere para los grupos de caché SWT y AWT. Con los grupos de caché SWT, cuando la aplicación finaliza una transacción que ha modificado uno o más grupos de caché, la transacción se ejecuta en la base de datos de Oracle primero y en IMDB Cache después. Esta técnica le permite a la base de datos de Oracle aplicar cualquier lógica solicitada que está relacionada con los datos antes de que sean ejecutados en IMDB Cache. Con los grupos de caché AWT, cuando la aplicación completa una transacción, la transacción se ejecuta en IMDB Cache y el control vuelve a la aplicación. Entonces, los cambios realizados por la transacción se propagan de manera asincrónica a la base de datos de Oracle.

- *Flush*. Esta operación se inicia por un pedido explícito de la aplicación y puede aplicarse tanto a grupos de caché como a instancias caché. Sólo se permite en grupos de caché o en instancias caché que tengan desactivada la opción de propagación. La operación actualiza los registros de la base de datos de Oracle con los valores de los registros en la caché. Esta operación es útil cuando se actualiza el mismo grupo de registros frecuentemente. En lugar de propagar el detalle de cada actualización, se envía y se aplica la imagen final de cada registro a la base de datos de Oracle.

Una aplicación es capaz de configurar una malla de caché con varios grupos de caché locales actualizables en distintos miembros de malla. La administración de la propagación de las actualizaciones desde los miembros de malla hacia la base de datos de Oracle está a cargo de IMDB Cache, pero se recomienda que los grupos de caché locales en los diferentes miembros de malla no se superpongan por si distintas actualizaciones de los mismos datos se realizan al mismo tiempo en nodos diferentes, lo cual podría tener como resultado valores de datos imprevisibles en el sistema.

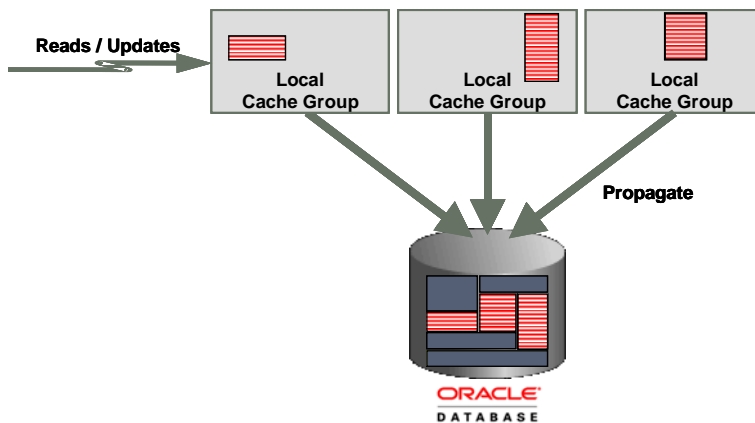


Figura 6. Propagación de actualizaciones para grupos de caché locales actualizables

4.4.3 Propagación de actualizaciones desde la base de datos de Oracle hacia IMDB Cache para grupos de caché locales

Para un grupo de caché local³ que se actualiza en la base de datos de Oracle, se encuentran disponibles los siguientes mecanismos a fin de mantener el contenido de la caché en sincronía con la base de datos de Oracle:

- *Refresh*. Se trata de una solicitud explícita de la aplicación para que actualice tanto un grupo de caché como una instancia caché específica. Es equivalente a una operación de descarga seguida de una operación de carga.
- *Full Autorefresh*. Con este mecanismo, la aplicación indica la frecuencia con la que deben realizarse las actualizaciones e IMDB Cache actualiza automáticamente el grupo de caché según los intervalos indicados por la aplicación.
- *Incremental Autorefresh*. A diferencia de Full Autorefresh, este mecanismo actualiza únicamente los registros que han sido modificados en la base de datos de Oracle a partir de la última actualización. Como con la Full Autorefresh, la aplicación deberá indicar la frecuencia de las actualizaciones e IMDB Cache realizará la actualización incremental automáticamente con esa frecuencia.

La actualización incremental automática puede combinarse con la antigüedad por tiempo a fin de mantener un intervalo móvil en la caché. Por ejemplo, una aplicación de servicio al cliente puede necesitar que se mantengan en la caché todos los incidentes registrados en los últimos cinco días. En ese caso, se puede especificar que el grupo de caché utilice una actualización incremental automática y una antigüedad por tiempo con una vida útil de cinco días. A medida que se insertan nuevos incidentes en la base de datos de Oracle, la actualización incremental automática propagará automáticamente a las tablas de caché en la memoria. Si esos incidentes se actualizan en la base de datos de Oracle, las actualizaciones se propagarán automáticamente a las tablas de caché en la memoria. Esos incidentes deberán tener una marca de fecha y hora mantenida por la aplicación. Cuando el valor de la marca de fecha y hora supere los cinco días de la fecha actual, el incidente asociado se eliminará automáticamente de la caché.

Las tres técnicas descritas anteriormente son útiles en distintas circunstancias. Supongamos que un grupo de caché necesita que se lo actualice solamente una vez por día a las 2 de la mañana, cuando la actividad en un sitio proveedor de contenidos es mínima. En ese caso, la mejor opción

³ Observe que los grupos de caché globales no pueden actualizarse en la base de datos de Oracle.

es una actualización completa. En cambio, si un grupo de caché necesita actualizarse cada cinco minutos, debería utilizarse el mecanismo de actualización incremental automática. Por último, un grupo de caché que únicamente precisa actualizaciones esporádicas, aunque a horarios imprevisibles que sólo conoce la aplicación, debería utilizar la opción Refresh.

Una aplicación es capaz de configurar una malla de caché con varios grupos de caché locales de sólo lectura en distintos miembros de malla. Los grupos de caché en los distintos miembros de malla pueden ser idénticos, inconexos o estar parcialmente superpuestos. IMDB Cache administrará la propagación de las actualizaciones desde la base de datos de Oracle hacia todos los miembros de malla.

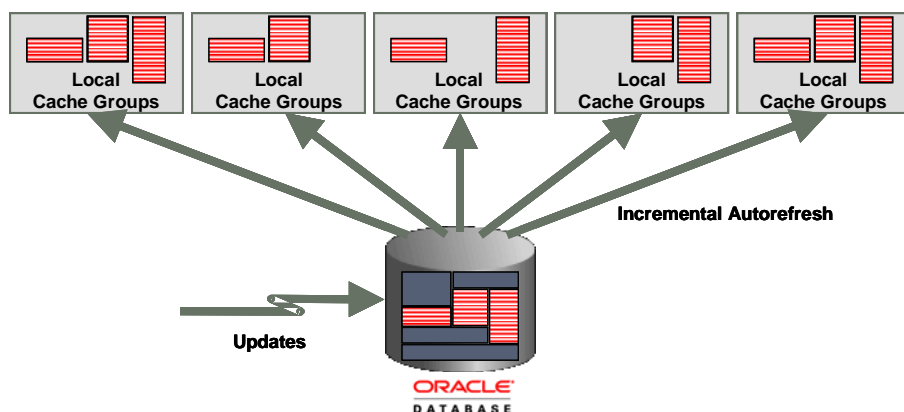


Figura 7. Actualización incremental automática de grupos de caché locales de sólo lectura

4.5 Alta disponibilidad

IMDB Cache permite una alta disponibilidad a través de la capa de aplicaciones y la capa del servidor de la base de datos.

Cuando se utiliza Oracle TimesTen exclusivamente como base de datos de registros, a diferencia de una base de datos caché de memoria de la base de datos de Oracle, se asegura la alta disponibilidad de sus datos a través de la replicación, diversas operaciones en línea y una cantidad de utilidades que permiten conmutación ante fallas, recuperación y actualizaciones en línea. La detección automática de fallas de la base de datos y las aplicaciones se encuentra disponible a través de la integración con Oracle Clusterware. De igual manera, la base de datos de Oracle permite una alta disponibilidad de sus datos a través de un grupo de características que incluye Oracle Real Application Clusters (RAC), Oracle Automatic Storage Management (ASM) y Oracle Data Guard. Además, el componente de replicación de IMDB Cache pone a su alcance numerosas características que aseguran la alta disponibilidad de los datos almacenados en caché y

la recuperación automática ante fallas que abarcan las capas de la aplicación y la base de datos de Oracle en la capa de la base de datos. A continuación, se describen estas características.

4.5.1 Manejo de fallas de un nodo de caché de memoria

Para protegerse contra las fallas de nodos de caché y asegurar la constante disponibilidad de los datos almacenados en caché, la replicación TimesTen brinda un manejo de la recuperación y las fallas de los nodos de caché. La configuración de la replicación Active Standby Pair con múltiples suscriptores de sólo lectura está diseñada para incluir una base de datos de Oracle como parte de la configuración y para brindar conmutación ante fallas y recuperación de nodos de caché.

Con Active Standby Pair, todas las actualizaciones se aplican siempre primero al nodo activo. Luego, se replican al nodo de reserva y, después, se replican del nodo de reserva hacia todos los nodos suscriptores de sólo lectura. Así, el nodo de reserva estará siempre un paso adelante de todos los nodos suscriptores de sólo lectura y, por lo tanto, si el nodo activo dejara de funcionar, no habría ninguna duda acerca de cuál de los nodos suscriptores deberá ser el nuevo nodo activo.

Grupos de caché de sólo lectura

La configuración de la replicación Active Standby Pair fue concebida para trabajar tanto con grupos de caché de sólo lectura como con grupos de caché con introducción sincrónica. Con grupos de caché de sólo lectura, las actualizaciones que se aplican en la base de datos de Oracle se propagan solamente hacia el nodo activo. Luego, la replicación de TimesTen propaga las actualizaciones a todos los otros nodos después de haber ido primero al nodo de reserva.

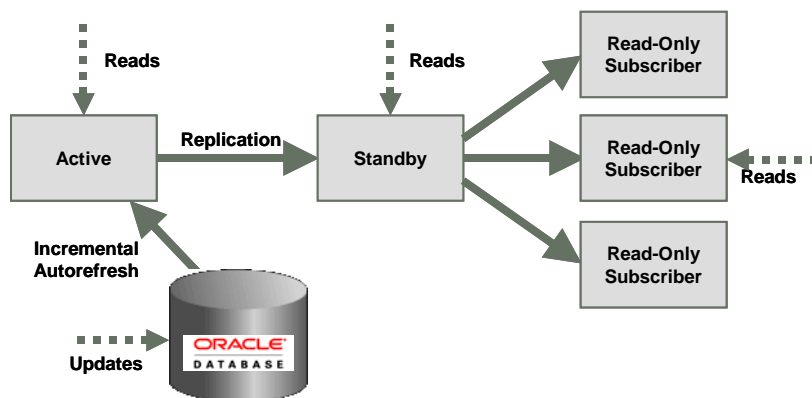


Figura 8. Grupos de caché de sólo lectura con configuración Active-Standby Pair

Si falla el nodo activo, el nodo de reserva se convierte en el nuevo nodo activo. A partir de ese momento, las actualizaciones de la base de datos de Oracle se propagan al nuevo nodo activo (anteriormente de reserva) y, después, se replican hacia los suscriptores de sólo lectura. Una vez que el antiguo nodo activo vuelve a estar en línea, se convierte en el nuevo nodo de reserva. La replicación de TimesTen conmuta la propagación de actualizaciones automáticamente desde el nodo activo al de reserva y se ocupa de la recuperación del nodo que falló.

De la misma forma, si falla el nodo de reserva, la replicación desde el nodo activo se redireccionará automáticamente hacia los nodos suscriptores de sólo lectura. Una vez que el nodo de reserva vuelve a estar en línea, la replicación hará que se ponga al día con todas las actualizaciones que omitió, antes de reanudar su función de nodo de reserva.

Grupos de caché con introducción sincrónica

En el caso de los grupos de caché con introducción sincrónica, las actualizaciones se aplican al nodo activo. Luego, se replican hacia el nodo de reserva. Desde allí, se propagan a la base de datos de Oracle y se replican hacia todos los nodos suscriptores de sólo lectura.

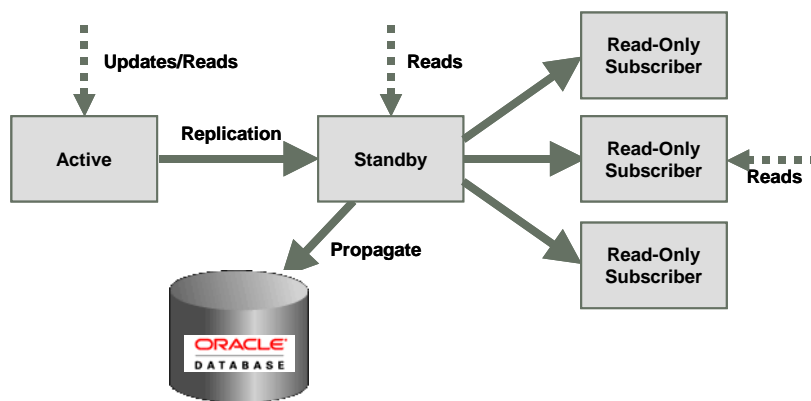


Figura 9. Grupos de caché con introducción que utilizan la configuración activo-de reserva.

Si falla el nodo activo, el nodo de reserva se convierte en el nuevo nodo activo. De allí en adelante, todas las actualizaciones deben enviarse al nuevo nodo activo, es decir, al anterior nodo de reserva. Las actualizaciones del nuevo nodo activo se propagarán a la base de datos de Oracle y se replicarán hacia los nodos suscriptores de sólo lectura. Una vez que el antiguo nodo activo vuelve a estar en línea, se convierte en el nuevo nodo de reserva. La replicación TimesTen maneja automáticamente la recuperación del nuevo nodo de reserva y le transfiere la responsabilidad de propagar las actualizaciones hacia el nuevo nodo de reserva.

Del mismo modo, si el nodo de reserva falla, la replicación desde el nodo activo se redireccionará automáticamente hacia los nodos suscriptores de sólo lectura, y el nodo activo comenzará a propagar las actualizaciones directamente a la base de datos de Oracle. Una vez que el nodo de reserva vuelva a estar en línea, la replicación se asegurará de que se ponga al día con todas las actualizaciones que omitió, antes de reanudar su función de nodo de reserva y hará que el nodo de reserva retome su rol de propagar las actualizaciones a la base de datos de Oracle y los suscriptores de sólo lectura.

4.5.2 Manejo de fallas en la base de datos de Oracle

En caso de que IMDB Cache no pueda acceder a la base de datos de Oracle por algún motivo, por ejemplo, por una falla en la red, una falla de hardware o una falla en Oracle Database, IMDB Cache está diseñada para poder salvar esas fallas. Las aplicaciones seguirán teniendo acceso a la caché de memoria. Además, en el caso de un grupo de caché con introducción asincrónica, las actualizaciones en la caché seguirán estando conectadas a Oracle TimesTen, de forma tal que cuando se recupere el acceso a la base de datos de Oracle, puedan propagarse las actualizaciones. Del mismo modo, los cambios a los grupos de caché de sólo lectura que se realizaron en la base de datos de Oracle pero que aún no se propagaron a la caché de memoria, permanecerán registrados en la base de datos de Oracle y se propagarán a la caché una vez restablecido el acceso a la base de datos de Oracle.

Asimismo, IMDB Cache aprovecha las ventajas de las características de alta disponibilidad de RAC. Una configuración RAC consiste en una sola base de datos física a la que tienen acceso varios nodos. La configuración del tiempo de ejecución de un solo nodo se denomina *instancia*. RAC brinda equilibrio de la carga, alta disponibilidad y coherencia entre los datos de todas las instancias.

IMDB Cache se recupera rápidamente de una falla de un nodo RAC, sin que se necesite la intervención del usuario. Para eso, IMDB Cache utiliza Transparent Application Failover (TAF) y Fast Application Notification (FAN) si están disponibles, según la versión de cliente y servidor Oracle y la configuración TAF. Si una instancia Oracle a la que se encuentra conectada IMDB Cache presenta fallas, la conexión se conmutará automáticamente hacia otra instancia. Si al momento de la falla se estaba realizando una operación de actualización, actualización completa o actualización incremental automática, se desharán los cambios que tuvieron lugar en la base de datos de memoria y se reiniciará la operación. Si al momento de la falla se estaba realizando una operación de propagación en un grupo de caché con introducción asincrónica, la transacción automáticamente deshará los cambios realizados en la base de datos de Oracle, de ser necesario, y reiniciará la operación de propagación.

Si la base de datos de Oracle se replica en una base de datos de reserva mediante Synchronous Data Guard, en caso de que ocurra una falla en la base de datos de Oracle activa, IMDB Cache

conmutará automáticamente a la base de datos de reserva, sin que se produzca pérdida alguna de datos.

5. Rendimiento

Para medir el rendimiento de IMDB Cache, desarrollamos un punto de referencia que simula una aplicación de registro de posición base (HLR) como la que utilizan las redes de telefonía móvil. El punto de referencia consta de siete transacciones, cada una de las cuales modela una operación típica ejecutada por un HLR, por ejemplo, establecer o eliminar el reenvío de una llamada o actualizar la información de un abonado de telefonía móvil.

Llevamos a cabo la comparación con el punto de referencia, en dos configuraciones distintas. En la primera configuración, la aplicación HLR se ejecutó contra la base de datos de Oracle, con la aplicación de referencia operando en un servidor y Oracle Database 10g en otro servidor. En la segunda configuración, agregamos IMDB Cache frente a la base de datos de Oracle. La aplicación HLR se vinculó directamente con la base de datos en caché TimesTen en un servidor, y Oracle Database 10g se ejecutó en el otro servidor. Los datos colocados en caché se almacenaron en grupos de caché con introducción asincrónica. Eso permitió que todas las actualizaciones con relación a los datos colocados en caché se propagaran automáticamente a la base de datos de Oracle.

La aplicación se implementó en Java, utilizando JDBC para el acceso a los datos. Los cuatro servidores tenían la misma configuración: 6 GB de RAM física y dos procesadores Intel Xeon de 2,4 GHz con tecnología hyper-threading en un entorno Oracle Enterprise Linux 5.2.

Se midió el tiempo de respuesta promedio para cada tipo de transacción ejecutada con la base de datos de Oracle y con IMDB Cache. El siguiente gráfico muestra una reducción significativa en el tiempo de respuesta de la aplicación con la utilización de IMDB Cache.

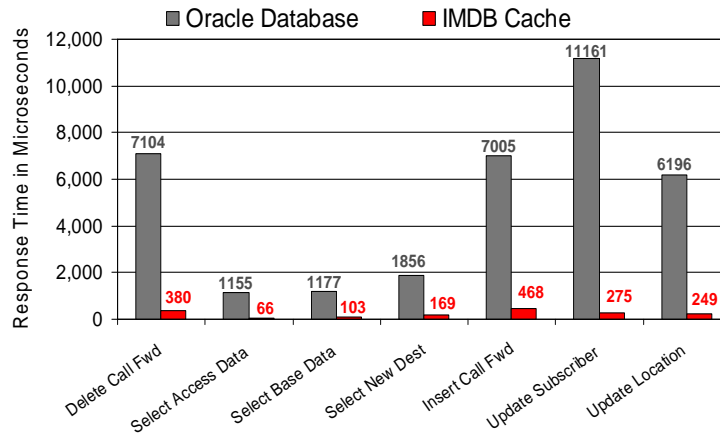


Figura 10. Comparación del tiempo de respuesta de la aplicación de referencia HLR.

También se midió el tiempo de procesamiento del total de las transacciones en ambas configuraciones. El siguiente gráfico muestra un aumento significativo en el tiempo de procesamiento de la aplicación con la utilización de IMDB Cache.

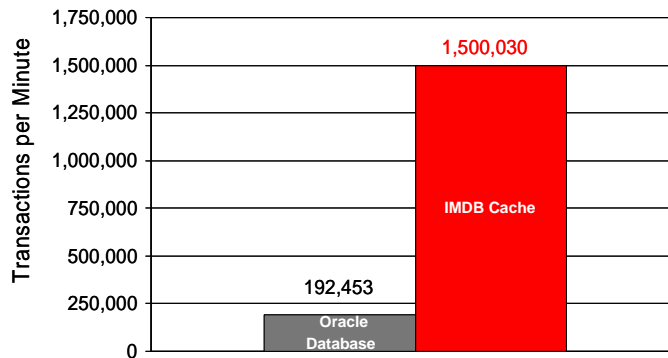


Figura 11. Comparación del tiempo de procesamiento de la aplicación de referencia HLR

Estos puntos de referencia ilustran los beneficios de la utilización de IMDB Cache. Como se ilustra en los gráficos, el tiempo de respuesta de la aplicación mejoró en un factor de 10 a 40, y el procesamiento general mejoró más de 7 veces. En términos generales, el índice de mejora varía en función del hardware y las plataformas.

6. Ejemplos

En esta sección se analizan algunos escenarios de almacenamiento en caché, así como también las configuraciones de IMDB Cache y los tipos de grupos de caché recomendados para cada uno de los escenarios. Cabe destacar que, si bien cada ejemplo se concentra en un tipo específico de grupo de caché, en una misma instancia de IMDB Cache puede haber diferentes tipos de grupos para adaptarse más adecuadamente a las necesidades de una aplicación.

6.1 Caché de sólo lectura

Los grupos de caché de sólo lectura con actualización incremental automática son ideales para el almacenamiento en caché de datos a los que se hace referencia frecuentemente.

Varias aplicaciones pueden beneficiarse con el uso de cachés de sólo lectura. La principal característica de esas aplicaciones es que tienen un conjunto de registros que se consultan una y otra vez. Los registros pueden actualizarse frecuentemente o no, pero el índice de lecturas respecto a escrituras es alto. Algunos de esos registros son las listas de precios para aplicaciones de compras en línea, los horarios de vuelos para aplicaciones de reserva aéreas y la disponibilidad de habitaciones en las aplicaciones de reservas de hoteles.

La mejor configuración de caché para ese tipo de información es el grupo de caché local de sólo lectura (gestionado por el sistema) con actualización incremental automática. Los datos se pueden almacenar en la base de datos interna. Las actualizaciones se propagarán automáticamente a la caché. La aplicación determina la frecuencia de la propagación, que deberá depender de la frecuencia de las actualizaciones internas y la vigencia de la información que necesita la aplicación.

Si la caché se va a implementar en varios miembros de una malla de caché, los grupos de caché locales se deberán definir a partir de los miembros de malla y cada miembro se actualizará directamente desde la base de datos interna.

Cabe destacar que, por lo general, una aplicación de compras en línea no necesitará actualizar con frecuencia las listas de precios almacenadas en caché, dado que no es habitual realizar cambios en las listas de precios. Por otro lado, una aplicación de seguimiento de vuelos, si bien se lee intensivamente, también necesita que se mantengan muy actualizados los estados de los vuelos almacenados en caché. Las actualizaciones se aplican mejor a la base de datos de Oracle con un intervalo de actualización incremental automática relativamente pequeño, por ejemplo, de cinco minutos, definido para los grupos de caché de sólo lectura. IMDB Cache propagará automáticamente todas las actualizaciones a los miembros de malla que incluyan los datos actualizados. Una aplicación que esté conectada a un miembro de malla puede configurar una

única conexión a IMDB Cache y utilizar la opción PassThrough para enviar todas las actualizaciones a la base de datos de Oracle mientras ejecuta todas las lecturas en la caché.

6.2 Caché de sólo lectura con intervalo móvil

Los grupos de caché de sólo lectura con actualización incremental automática y antigüedad por tiempo son ideales para la información a la que se hace referencia frecuentemente dentro de un intervalo móvil.

En muchos casos, los datos de sólo lectura que necesita una aplicación representan información con un componente temporal, en la que se accede con mayor frecuencia a los datos más recientes que a los más antiguos. En este tipo de aplicaciones específicas, se generan constantemente datos nuevos y los más antiguos se vuelven menos valiosos. Por lo tanto, se puede pensar en un intervalo temporal de extensión fija que avanza constantemente con información que aumenta en uno de los extremos del intervalo y decae en el otro. La aplicación se ocupa sólo de la información que se encuentra en ese intervalo, que se conoce generalmente como "intervalo móvil".

Algunos ejemplos de aplicaciones que pueden llegar a necesitar un intervalo móvil son una aplicación de comercio bursátil que puede necesitar el registro de los últimos tres días de operaciones, o una aplicación de noticias que necesite los artículos de las últimas 24 horas.

Para almacenar en caché la información de un intervalo móvil, es conveniente que los datos nuevos se incluyan automáticamente en la caché y los antiguos caduquen también automáticamente. También es necesario que los datos que ya se encuentren en la caché se actualicen automáticamente en caso de que se realicen modificaciones internas. La mejor configuración de caché para ese tipo de información es el grupo de caché de sólo lectura (gestionado por el sistema) con actualización incremental automática y antigüedad por tiempo.

Como en el ejemplo anterior, si la caché se va a implementar en varios miembros de una malla de caché, los grupos de caché locales se deberán definir a partir de los miembros de malla y cada miembro se actualizará directamente desde la base de datos de Oracle.

6.3 Caché actualizable

Los grupos de caché con introducción asincrónica son ideales para cachés actualizables.

Algunas aplicaciones exigen una actualización inmediata y en tiempo real de la información almacenada en caché, con la posibilidad de que las actualizaciones se propaguen a una base de datos de Oracle. Por ejemplo, una aplicación que administra y brinda servicios de telefonía y autentica el acceso a esos servicios, por lo general, almacenará en caché la información de los abonados en IMDB Cache. Los cambios que se realizan en el servicio de un usuario se deben reflejar inmediatamente en la caché y propagarse a la base de datos interna.

La mejor configuración para ese tipo de datos es un grupo de caché con introducción asincrónica.

Si la cantidad de abonados es lo bastante grande como para requerir que la aplicación se implemente en varios miembros de una malla de caché, los grupos de caché locales se deberán definir en los miembros de malla, en los que el grupo de caché de cada miembro esté a cargo de un subconjunto de abonados y sin superposición entre los subconjuntos que pertenecen a miembros de diferentes mallas. Por ejemplo, los miembros se pueden dividir por código de área.

6.4 Caché dinámica actualizable

Los grupos de caché globales con introducción asincrónica, carga dinámica y antigüedad por uso son ideales para las cachés dinámicas actualizables.

Para algunas aplicaciones, el acceso a la información activa debe ser muy rápido. Pero el conjunto de datos activos varía con el tiempo y constituye un subconjunto de datos mucho mayor que, debido a su tamaño, no se puede contener por completo en una caché. Es necesario que la información activa se incluya en la caché en función de la demanda y que el contenido de la caché sea dinámico para que los datos activos puedan reemplazar a los obsoletos.

Un ejemplo de aplicaciones de este tipo es un centro de llamadas que administra un gran volumen de sesiones de clientes al mismo tiempo. La aplicación generalmente se implementará en varios nodos de servidores de aplicaciones. Los clientes que se contactan con el centro de llamadas se derivan automáticamente a un nodo de servidor de aplicaciones disponible. Idealmente, en ese nodo debería estar el perfil de cliente correspondiente.

La mejor configuración para este caso son los grupos de caché globales con introducción asincrónica, carga dinámica y antigüedad por uso, y que además tengan un miembro de malla configurado en cada nodo del servidor de aplicaciones.

Con esta configuración, cuando se deriva un cliente a un nodo disponible, el perfil de cliente pertinente se carga dinámicamente desde la base de datos de Oracle al miembro de malla. Cuando el cliente finaliza la llamada, los cambios realizados a su perfil se propagan de IMDB Cache a la base de datos de Oracle. La antigüedad LRU eliminará automáticamente los perfiles de los usuarios inactivos de IMDB Cache. Si, poco después, el mismo cliente se contacta nuevamente con el centro de llamadas y es derivado a un nodo distinto, el perfil del cliente se cargará dinámicamente en el nuevo nodo, ya sea desde la base de datos de Oracle o desde el miembro de malla donde se había cargado previamente, dependiendo del lugar en el que se encuentre la copia más reciente. IMDB Cache determina dónde se encuentra la copia más reciente. También administra las actualizaciones de datos que se producen al mismo dentro de la malla.

Toda la información de los clientes se almacena en la base de datos de Oracle. La base de datos de Oracle es más grande que todas las de IMDB Cache combinadas, y acceden mejor a ella las aplicaciones que no exigen que IMDB Cache proporcione rendimiento en tiempo real, pero sí necesitan acceso a grandes cantidades de información. Entre esas aplicaciones, se encuentran las de facturación y las de extracción de datos.

A medida que crecen la base de clientes y la demanda de brindar servicios a más clientes simultáneamente, el centro de llamadas puede decidir implementar nodos de servidores de aplicaciones adicionales. Pueden incorporarse nuevos miembros de IMDB Cache a la malla de IMDB Cache, sin interrumpir las continuas solicitudes en la malla. De manera similar, las fallas o la eliminación de nodos individuales no interrumpen las operaciones del resto de la malla.

6.5 Caché de captura de datos con tasa de ingreso irregular

Los grupos de caché con introducción asincrónica y antigüedad por uso son ideales para capturar datos con tasas de ingreso irregulares.

Existe un tipo de aplicaciones en los que la información se genera a muy alta velocidad en algunos períodos y a velocidad moderada en otros. Durante los períodos de gran actividad, la base de datos interna a menudo no puede estar a la altura del elevado procesamiento que demanda la aplicación. Esas aplicaciones se pueden beneficiar con el uso de una caché que, de hecho, "mitiga" el ingreso de datos recién generados.

Por ejemplo, en una aplicación de cotizaciones, la tasa de ingreso de nuevos valores variará ampliamente en función del tiempo. Será particularmente alta durante las aperturas y los cierres del mercado, y menor en otros períodos. En el punto crítico, la tasa de ingresos normalmente no se puede administrar con una base de datos que funciona en un disco, pero es posible sustentarla con IMDB Cache.

La mejor configuración de caché para ese tipo de información son los grupos de caché (gestionados por el sistema) con introducción asincrónica y antigüedad por uso. Los ingresos en los grupos de caché con introducción se propagan automáticamente a la base de datos interna de Oracle. Y la antigüedad por uso eliminará automáticamente la información de la caché de memoria para liberar espacio.

6.6 Caché de captura de datos con tasa de ingreso constantemente alta

Los grupos de caché con introducción asincrónica y antigüedad por uso, asociados a tablas TimesTen convencionales con antigüedad por uso, son ideales para capturar datos con una tasa de ingreso constantemente alta.

Existe otro tipo de aplicaciones en las que la nueva información también se genera a alta velocidad, pero la tasa de ingreso no necesariamente decae. El almacenamiento temporal en

caché de datos cuya tasa de ingreso es muy alta para que la absorba la base de datos interna no soluciona el problema si la tasa de ingreso no decae, dado que no hay un paréntesis en el que la base de datos interna pueda ponerse al día. Sin embargo, suele darse el caso de que, para aplicaciones de ese tipo, los datos generados se pueden agregar en forma más condensada antes de que se los almacene permanentemente en la base de datos interna. También es común que esas aplicaciones analicen los datos que recopilan en tiempo real a fin de detectar patrones anómalos o de interés.

Un ejemplo de esas aplicaciones son las que recopilan datos a partir de sensores o lectoras de identificación por radiofrecuencia (RFID). A menudo la información es repetitiva, se puede agregar fácilmente y exige un análisis en tiempo real.

La mejor configuración para las aplicaciones de ese tipo es el ingreso de la información a medida que va llegando en una o más tablas que se administran sólo por medio de Oracle TimesTen. Es decir, no hay imagen de esa información en la base de datos interna. Las tablas no almacenadas en caché mediante el uso exclusivo de tablas TimesTen se pueden configurar con antigüedad por uso. Una vez que la aplicación agrega la información, ésta se puede ingresar a la caché en un grupo de caché (gestionado por el sistema) con introducción asincrónica y antigüedad por uso. IMDB Cache propagará automáticamente todos los datos agregados a la base de datos interna. Dado que tanto las tablas TimesTen como las de caché se configuran con antigüedad por uso, los registros que menos se hayan utilizado recientemente se eliminarán en forma automática a fin de generar espacio en la caché para los registros nuevos.

6.7 Caché actualizable gestionada por el usuario

Los grupos de caché gestionados por el usuario con registro de cambios explícito son ideales para aplicaciones que reciben actualizaciones frecuentes, pero realizan transacciones comerciales esporádicas.

Algunas aplicaciones necesitan ejecutar diversas actualizaciones en la caché para obtener un mejor rendimiento, pero también necesitan mantener un registro permanente de la transacción final en la base de datos de Oracle. Un ejemplo de este tipo de aplicaciones son las del comercio por medios electrónicos, en las que la aplicación puede mantener una cantidad de carros de compra para usuarios activos. Los carros de compra se actualizarán varias veces en la caché. Dado que no tienen mucho valor, no es necesario que esas actualizaciones se propaguen a la base de datos de Oracle. Sin embargo, cuando un usuario realiza una compra, es necesario que la transacción se registre permanentemente en la base de datos de Oracle.

La mejor configuración para ese tipo de datos es un grupo de caché actualizable gestionado por el usuario, en el que la aplicación emite solicitudes de registro de cambios explícitas cada vez que necesita registrar una transacción en la base de datos de Oracle. Esta configuración se puede

asociar con la antigüedad por uso, de manera que los carros de compra abandonados se eliminen automáticamente de la caché.

6.8 Caché de sólo lectura, dinámica y distribuida

Las tablas de sólo lectura con carga dinámica y antigüedad por uso son ideales para una caché de sólo lectura, dinámica y distribuida.

En algunos casos, una aplicación se puede distribuir en muchos nodos para manejar una tasa de procesamiento que no se puede controlar mediante un solo nodo. El conjunto de datos activos que necesita la aplicación es dinámico y, en un momento determinado, constituye un subconjunto mucho menor que todo el conjunto de datos. Un ejemplo puede ser una aplicación comercial en la que la información activa está constituida por el perfil de comerciantes activos.

La mejor solución para esos datos es configurar la caché de memoria en cada nodo con un grupo de caché de sólo lectura en el mismo conjunto de tablas en la base de datos de Oracle, y utilizar la carga dinámica y la antigüedad por uso para esas tablas de caché. Lo que sucederá es que los perfiles de usuarios se cargarán automáticamente en cada nodo cuando se los necesite y se eliminarán de las cachés cuando ya no sean necesarios, con el fin de generar espacio para los perfiles necesarios.

7. Conclusión

Oracle In-Memory Database Cache le permite mejorar el tiempo de respuesta de transacciones de las aplicaciones mediante el almacenamiento en caché de subconjuntos de tablas de rendimiento crítico y fragmentos de tabla desde una base de datos de Oracle a la capa de aplicaciones. A diferencia de los mecanismos simples de caché de resultados, las aplicaciones pueden ejecutar comandos SQL y PL/SQL en los datos almacenados en caché, ya que las tablas de caché se administran como tablas convencionales de bases de datos relacionales en TimesTen In-Memory Database. Distintas aplicaciones pueden compartir las cachés. Se pueden aplicar actualizaciones a las cachés, las cuales se mantienen en coherencia con la base de datos de Oracle. El almacenamiento de datos en caché utilizando IMDB Cache es superior al de otras técnicas porque ofrece una completa funcionalidad relacional, una escalabilidad incremental asociada con una transparencia de ubicación, un rendimiento superior, un mantenimiento automático de datos coherente con la base de datos de Oracle y una alta disponibilidad entre capas para aplicaciones que funcionan en la capa de aplicaciones.

Al acercar los datos a la aplicación y procesar las consultas en una base de datos de memoria, Oracle In-Memory Database Cache reduce considerablemente el tiempo de respuesta. Dado que descarga parte del trabajo de procesamiento de datos del servidor de base de datos de Oracle, se

reduce significativamente el procesamiento general de las aplicaciones sin interferir con el manejo y la administración centralizada de la base de datos interna.

8. Referencias

1. Rendimiento extremo mediante la tecnología de base de datos de memoria TimesTen.
Documento técnico de Oracle: julio de 2009.



Uso de Oracle In-Memory Database Cache para
acelerar la base de datos de Oracle
Julio de 2009

Oracle Corporation
Sede mundial
500 Oracle Parkway
Redwood Shores, CA 94065
EE. UU.

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2008, 2009, Oracle Corporation y/o sus subsidiarias. Todos los derechos reservados. Este documento se proporciona exclusivamente con fines informativos y sus contenidos están sujetos a cambio sin previo aviso. No se garantiza que el presente documento esté libre de errores, ni tampoco está sujeto a otras garantías y condiciones, ya sean expresadas oralmente o implícitas en la legislación, incluidas garantías y condiciones implícitas de comerciabilidad o adecuación para un propósito en particular. Oracle se exime específicamente de cualquier responsabilidad respecto del presente documento y tampoco se establece relación contractual directa o indirecta alguna mediante el mismo. El presente documento no podrá reproducirse, ni transmitirse de forma ni por medio alguno, sea éste electrónico o mecánico, para finalidad alguna, sin el expreso consentimiento escrito de Oracle Corporation.

Oracle es una marca comercial registrada de Oracle Corporation y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.