

Um artigo técnico da Oracle
Julho de 2009

Usando o Oracle In-Memory Database Cache para acelerar o banco de dados Oracle

1. Introdução	2
2. Armazenamento em cache na camada de aplicativos.....	4
3. O Oracle TimesTen In-Memory Database	6
3.1 Desempenho do Oracle TimesTen	9
4. Armazenamento em cache usando o Oracle In-Memory Database Cache	9
4.1 Definindo o conteúdo de um cache.....	11
4.2 Carregando dados e gerenciando o cache	12
4.3 Compartilhando dados através de uma Grade de cache.....	14
4.4 Mantendo a consistência de dados.....	14
4.5 Alta disponibilidade	21
5. Desempenho	24
6. Exemplos	26
6.1 Cache somente para leitura	26
6.2 Cache de "janela deslizante" somente para leitura.....	27
6.3 Cache que pode ser atualizado	28
6.4 Cache dinâmico que pode ser atualizado	28
6.5 Cache de captura de dados com Taxa de chegada desigual	29
6.6 Cache de captura de dados com Taxa de chegada alta constante	30
6.7 Cache gerenciado pelo usuário que pode ser atualizado	31
6.8 Cache distribuído dinâmico somente para leitura	31
7. Conclusão	31
8. Referências	32

1. Introdução

O Oracle In-Memory Database Cache acelera os processos da empresa, proporciona inteligência comercial em tempo real e facilita a personalização de aplicativos de contato direto com o cliente.

O *Oracle In-Memory Database Cache (IMDB Cache)* é uma opção de produto do banco de dados Oracle ideal para armazenar em cache subconjuntos de desempenho crítico de um banco de dados Oracle na camada de aplicativos. Ao usar o IMDB Cache, o tempo de resposta e o throughput do aplicativo são melhorados. O IMDB Cache é composto por três componentes principais de tecnologia – o Oracle TimesTen In-Memory Database (TimesTen) para o gerenciamento de dados em tempo real na camada de aplicativos; a tecnologia de armazenamento em cache para armazenar tabelas acessadas com frequência de um servidor de banco de dados Oracle para a camada de aplicativos e manter a consistência dos dados armazenados; e um componente de replicação de dados transacionais para garantir a alta disponibilidade entre camadas.

O TimesTen é um banco de dados relacional com memória otimizada que fornece tempo de resposta muito baixo e throughput muito alto a sistemas com desempenho crítico. Ele foi projetado para ser executado na camada de aplicativos, próximo aos aplicativos, e, opcionalmente, no processo com os aplicativos. Um banco de dados TimesTen pode ser usado como um banco de dados de registro e/ou como um armazenamento em cache de um banco de dados Oracle.

Os aplicativos podem criar e gerenciar tabelas de banco de dados no TimesTen ou armazenar em cache subconjuntos de um banco de dados Oracle acessados com frequência no IMDB Cache. As tabelas armazenadas e as tabelas não armazenadas podem coexistir no mesmo banco de dados in-memory e serem todas persistentes e recuperáveis. As consultas e atualizações dos dados armazenados e não armazenados são realizadas por aplicativos através do SQL92 ou PL/SQL usando ODBC, JDBC, a Oracle Call Interface (OCI) ou TTCclasses, bem como Pro*C.

Grades de cache estão disponíveis para escalabilidade horizontal em desempenho e capacidade e consistem em uma coleção de IMDB Caches que gerenciam coletivamente os dados armazenados em cache de um aplicativo. Os dados armazenados em cache são distribuídos entre os membros da grade e a Grade de cache fornece aplicativos com transparência de local, fazendo a agregação dos dados armazenados de forma eficiente em todos os membros da grade disponíveis para o aplicativo. As Grades de cache permitem a escalabilidade incremental através da adição (e remoção) on-line de membros da grade. Elas mantêm a consistência dos dados armazenados em cache entre os membros da Grade de cache e o banco de dados Oracle.

O IMDB Cache gerencia a disponibilidade dos dados através da camada de aplicativos e a camada do servidor do banco de dados. Ele garante alta disponibilidade e nenhuma perda de transações independentemente de onde ocorrer uma falha. Seja a ocorrência da falha em um dos nós do cache, um dos nós do Oracle RAC, no nível da rede ou até mesmo em um cluster inteiro de RAC, são garantidas a alta disponibilidade e nenhuma perda de transações.

O TimesTen e o IMDB Cache têm um histórico comprovado em implantações de produção em empresas em tempo real e setores críticos que incluem serviços de redes de telecomunicações, sistemas de suporte operacional, centros de contato, companhias aéreas e sistemas de reserva, sistemas de controle e comando e mercado de ações. Centenas de empresas no mundo todo usam o TimesTen e o IMDB Cache em aplicativos de produção, incluindo a Alcatel-Lucent, Amdocs, Aspect, Avaya, Bombay Stock Exchange, Bridgewater Systems, BroadSoft, Cisco, Deutsche Börse, Ericsson, JP Morgan, NEC, NYFIX, Smart Communications e Sprint.

2. Armazenamento em cache na camada de aplicativos

O armazenamento em cache na camada de aplicativos é normalmente usado para melhorar a latência do acesso aos dados e reduzir a carga de trabalho no banco de dados de back-end.

Diversas técnicas de armazenamento em cache têm sido desenvolvidas para melhorar o desempenho do acesso ao banco de dados ou para reduzir a contenção em servidores de banco de dados de back-end. O rápido tempo de resposta é especialmente importante para aplicativos em tempo real e aplicativos de contato direto com o cliente. Além disso, reduzir a carga de trabalho no banco de dados de back-end é importante para aplicativos com uma comunidade de usuários em constante crescimento como serviços de softwares hospedados, sites de comércio eletrônico ou serviços de telecomunicações.

Existem diversas opções sobre quais informações armazenar em cache e onde armazená-las, cada uma delas oferecendo vantagens e compensações. Algumas das técnicas de armazenamento em cache desenvolvidas incluem:

- *Cache de resultados de consultas.* Isso é normalmente feito na camada de aplicativos e é gerenciado por softwares especiais que ocultam do aplicativo a presença do armazenamento em cache. Neste caso, o software de armazenamento em cache salva automaticamente os resultados das consultas que são enviadas para o sistema de banco de dados. Uma solicitação do cache é reconhecida e gerenciada pelo cache se uma consulta é uma correspondência exata de uma consulta enviada anteriormente, incluindo valores e parâmetros idênticos. A vantagem deste tipo de armazenamento em cache é que ele é simples e atende a necessidade de casos de acesso onde existe a probabilidade de a mesma consulta ser enviada diversas vezes. Entretanto, ele possui um escopo limitado, pois não consegue lidar com o processamento de consultas no conteúdo do cache.
- *Cache de ferramentas de mapeamento objeto-relacionais.* As ferramentas de mapeamento objeto-relacionais (ferramentas de mapeamento O/R) ocultam os bancos de dados relacionais de programadores orientados a objetos fornecendo o mapeamento transparente entre objetos e dados relacionais. Uma vez que os dados relacionais são mapeados para uma representação de objeto, eles podem ser armazenados pela ferramenta de mapeamento O/R até que não sejam mais necessários ou se tornem obsoletos. O armazenamento em cache através de ferramentas de mapeamento O/R é uma técnica comum para evitar o oneroso mapeamento entre o modelo de objeto da linguagem de programação e o modelo relacional do banco de dados.
- *Cache de objetos.* A expressão "armazenamento em cache" pode parecer um pouco inadequada neste caso porque os objetos armazenados nesse cache não são necessariamente subconjuntos de objetos que estão armazenados em outro lugar. Esses "armazenamentos em

cache” são repositórios de objetos que são independentes de sua origem. Eles normalmente não são transparentes para os aplicativos. Os aplicativos executam as operações “put”, “get”, “insert” e “delete” com os objetos a partir de seus armazenamentos em cache. Existem poucos produtos no mercado que oferecem esse tipo de armazenamento em cache e eles variam em relação ao nível de funcionalidade que suportam. Os armazenamentos em cache podem residir apenas na memória ou podem ser copiados para o disco ou outro sistema de gerenciamento de dados. Alguns produtos fornecem controle de simultaneidade, alguns fornecem distribuição transparente sobre múltiplos nós em uma rede, e outros fornecem alta disponibilidade.

O Oracle In-Memory Database Cache tem total funcionalidade relacional e SQL, manutenção automática de consistência de dados com o banco de dados Oracle e desempenho em tempo real.

O Oracle In-Memory Database Cache (IMDB Cache) usa uma abordagem exclusiva ao permitir o armazenamento em cache de tabelas e fragmentos de tabelas a partir de um banco de dados Oracle para a camada de aplicativos. Os fragmentos de tabela são descritos através de uma sintaxe SQL estendida e são armazenados em um Oracle TimesTen In-Memory Database (TimesTen). Os aplicativos leem e atualizam os dados armazenados em cache usando SQL, PL/SQL ou Pro*C e o IMDB Cache propaga atualizações automaticamente a partir de um banco de dados Oracle para o armazenamento em cache e vice-versa.

Uma coleção de IMDB Caches pode ser configurada como uma Grade de cache. Os dados armazenados em cache são distribuídos entre os membros da grade e a Grade de cache fornece aplicativos com transparência de local e controle de simultaneidade, fazendo a agregação de todos os dados armazenados de forma eficiente em todos os membros da grade disponíveis para os aplicativos. Conforme a necessidade de desempenho ou capacidade de um aplicativo aumenta, outros nós podem ser adicionados à Grade de cache sem interrupção do serviço. Desta forma, o IMDB Cache oferece aos aplicativos a generalidade e a funcionalidade completa de um banco de dados relacional, escalabilidade incremental associada à transparência local, manutenção automática da consistência de cache com o banco de dados Oracle e desempenho em tempo real de um banco de dados in-memory.

A abordagem do IMDB Cache tem duas vantagens principais que contribuem para melhorar o desempenho geral. Primeiro, os aplicativos que usam o IMDB Cache obtêm tempo de resposta significativamente reduzido e maior throughput devido à arquitetura in-memory do TimesTen e a eliminação da comunicação entre a camada de aplicativos e o servidor de banco de dados. Em segundo lugar, esta abordagem reduz a carga de trabalho no banco de dados de back-end melhorando assim o throughput geral de todos os aplicativos.

A capacidade de fornecer todas as vantagens de bancos de dados relacionais, associada ao desempenho em tempo real, escalabilidade incremental e gerenciamento automático do cache, é

exclusiva do IMDB Cache. É ideal para o armazenamento em cache de subconjuntos de um banco de dados Oracle com desempenho crítico, permitindo tanto a leitura como a atualização dos dados armazenados em cache, e gerenciando de forma automática a consistência de dados.

As próximas seções fornecerão uma breve introdução ao Oracle TimesTen In-Memory Database (mais detalhes podem ser obtidos em [1]), uma descrição de como os dados são armazenados em cache e gerenciados pelo Oracle In-Memory Database Cache e alguns casos ilustrativos de armazenamento em cache.

3. O Oracle TimesTen In-Memory Database

O TimesTen In-Memory Database fornece acesso transacional a dados e funcionalidade relacional através de APIs padrão.

O Oracle TimesTen In-Memory Database é um banco de dados relacional com memória otimizada que suporta SQL92 e PL/SQL através de APIs de ODBC, JDBC, Oracle Call Interface (OCI) e ITClasses¹, bem como através do Pro*C/C++. Ao oferecer suporte a interfaces padrão e interfaces conhecidas da Oracle, o TimesTen garante a facilidade de adoção por aplicativos já existentes.

Embora o TimesTen opere em dados que estão na memória principal, os bancos de dados TimesTen são persistentes e recuperáveis em caso de falha de software, hardware ou falta de energia. A durabilidade é garantida através de pontos de verificação e registro no disco. Os aplicativos podem escolher as propriedades ACID de suas transações, mas opções mais flexíveis também estão disponíveis para obter maior desempenho. O TimesTen fornece um otimizador de consultas com base em custo e os aplicativos podem visualizar e influenciar os planos de consultas. O banco de dados TimesTen está disponível como uma biblioteca que pode ser vinculada por aplicativos bem como através de uma opção cliente/servidor. Quando o TimesTen é acessado através da opção cliente/servidor, cada solicitação ao TimesTen resulta na sobrecarga de comunicação entre processos mesmo se o aplicativo e o servidor do TimesTen estiverem sendo executados na mesma máquina. Por outro lado, quando o TimesTen é vinculado ao aplicativo, as solicitações ao TimesTen nada mais são que chamadas locais que resultam em uma insignificante sobrecarga e quaisquer transferências de dados entre o aplicativo e o TimesTen

¹ As Classes de interface C++ do TimesTen (ITClasses) são uma biblioteca de classes C++ que fornecem wrappers para as funcionalidades ODBC mais conhecidas. É mais fácil de usar que o ODBC e incentiva as práticas recomendadas ao mesmo tempo em que mantém rápido desempenho.

nada mais são que operações de cópias de memória de baixo custo. A alta disponibilidade é proporcionada através da replicação. Diversos utilitários também estão disponíveis, incluindo um utilitário interativo de SQL, uma ferramenta gráfica para desenvolvimento de banco de dados e configuração do cache, backup e restauração on-line e carregamento em massa. As operações de banco de dados também estão disponíveis através de APIs de programas.

Uma cópia do banco de dados reside na memória principal em tempo de execução. Ela é gerenciada em um segmento de memória compartilhado que é acessado por todos os processos conectados a esse banco de dados. A Figura 1 mostra a arquitetura de um sistema do TimesTen In-Memory Database.

As estruturas de dados e os algoritmos do Oracle TimesTen In-Memory Database são otimizados em relação à residência dos dados na memória.

As estruturas de dados e os algoritmos de acesso do TimesTen exploram a residência do banco de dados na memória para obter um desempenho inovador. Comparado a um banco de dados com base em disco totalmente armazenado em cache, a arquitetura de memória otimizada do TimesTen usa muito menos ciclos de CPU, porque a sobrecarga para gerenciar os buffers e a conta de memória para múltiplos locais de dados (disco e memória) é eliminada.

O desempenho de memória otimizada do Oracle TimesTen é complementado pela funcionalidade que suporta propriedades transacionais, mecanismos de persistência e recuperação de falhas no sistema. Uma variedade de opções está disponível para bloqueio, isolamento e registro de múltiplos usuários, acomodando inúmeros cenários de aplicações desde armazenamentos em cache de pesquisas transitórias a sistemas transacionais importantes de mercado financeiro e cobrança em telecomunicações.

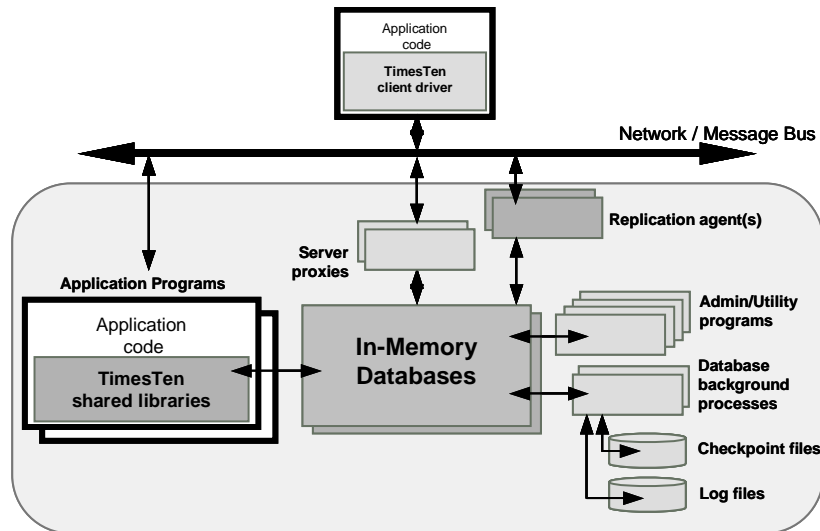


Figura 1. Arquitetura do TimesTen

Os bancos de dados do TimesTen são persistentes e recuperáveis.

A persistência é obtida no TimesTen registrando as alterações de transações confirmadas em disco e atualizando periodicamente uma imagem de disco do banco de dados através de pontos de verificação. O intervalo de tempo para gravação do disco para o registro pode ser configurado por aplicativo, tanto síncrono ao final da transação ou adiado, resultando em maior desempenho. Muitas situações favorecem um throughput maior no registro síncrono, especialmente quando o valor monetário de uma transação de dados tem vida curta, como ao rastrear o local de telefones móveis em uma rede que se comunica com seus locais de célula a cada poucos segundos.

O TimesTen permite que os aplicativos rastreiem as alterações em tabelas específicas. Isso é útil em ambientes onde os aplicativos podem ter comportamentos diferentes em função de determinados eventos. Por exemplo, um aplicativo pode querer saber quando o preço de uma determinada ação ultrapassou um limite predeterminado. Esse recurso de notificação de alteração é especialmente útil ao permitir que o rastreamento de alterações não seja realizado somente em tabelas de base, mas também em visualizações materializadas.

3.1 Desempenho do Oracle TimesTen

Tempos de resposta muito baixos não podem ser obtidos através da adição de hardware. O TimesTen fornece níveis baixíssimos de latência devido à sua arquitetura exclusiva.

O TimesTen pode atingir tempos de resposta da ordem de microssegundos devido à sua arquitetura exclusiva. Com o TimesTen, uma transação que lê um registro do banco de dados pode levar menos de 5 microssegundos e as transações que atualizam ou inserem um registro podem levar menos de 15 microssegundos.

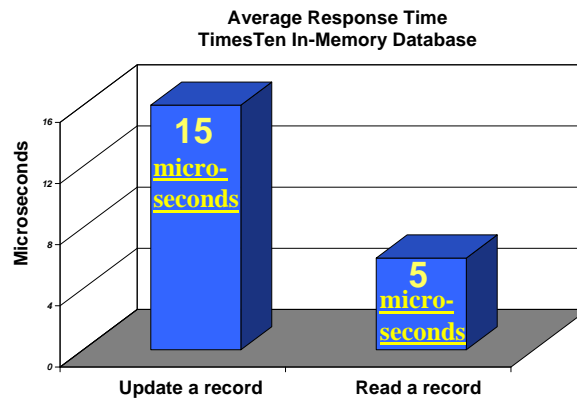


Figura 2. Tempo de resposta do TimesTen

A Figura 2 mostra os tempos de resposta de um aplicativo executando transações de leitura e atualização em um sistema com 2 CPUs Intel E5450 (8 vias a 3 GHz) executando o Oracle Enterprise Linux 5.2.

4. Armazenamento em cache usando o Oracle In-Memory Database Cache

Um IMDB Cache contém subconjuntos de tabelas de um banco de dados Oracle.

O IMDB Cache possibilita o armazenamento em cache de subconjuntos de tabelas de um banco de dados Oracle na camada de aplicativos. As tabelas armazenadas em cache podem ser atualizadas e o IMDB Cache sincroniza os dados entre o banco de dados Oracle e o cache.

O mecanismo de banco de dados que gerencia os dados armazenados em cache é o Oracle TimesTen In-Memory Database. Ele é ampliado através da capacidade de carregar e sincronizar dados armazenados em cache. Um dos processos de segundo plano associados ao IMDB Cache

é o Cache Agent, que gerencia parte desta sincronização. A Figura 3 mostra a arquitetura de um IMDB Cache.

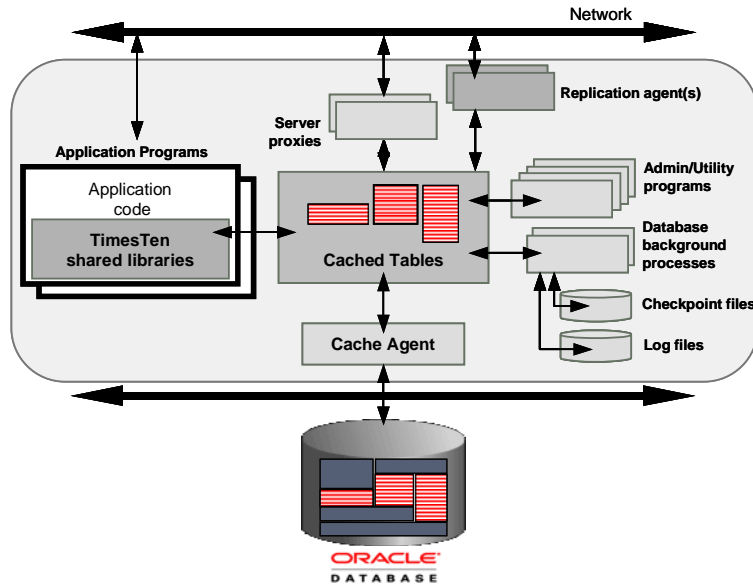


Figura 3. Arquitetura do IMDB Cache

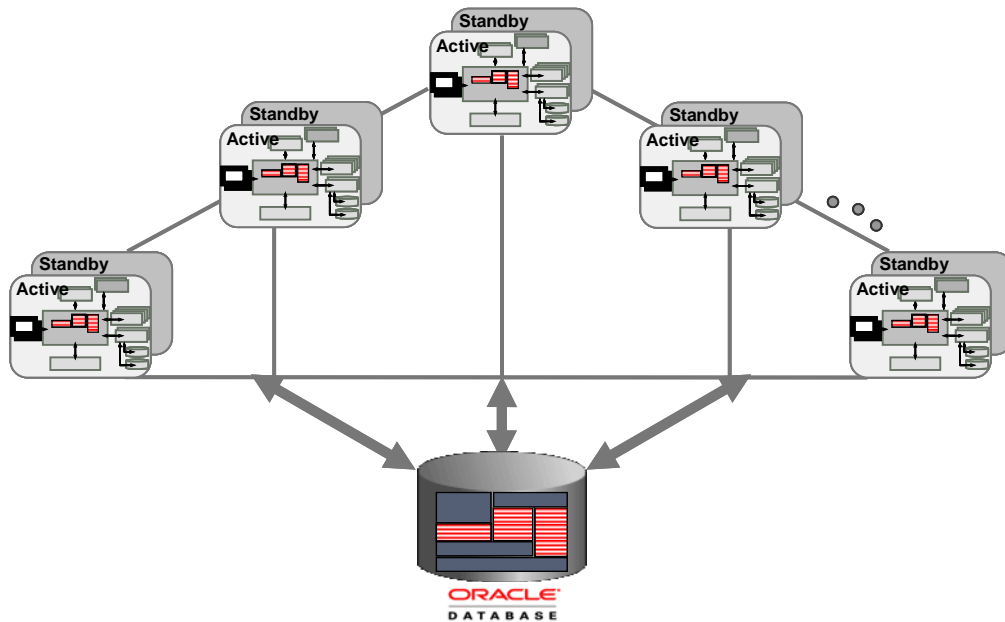


Figura 4. Grade de cache com cinco membros de grade replicados

Uma *Grade de cache* é uma coleção de IMDB Caches que gerenciam coletivamente os dados de aplicativo. Uma grade de cache é composta de um ou mais membros da grade, cada um suportado por um IMDB Cache. Os membros da grade armazenam em cache as tabelas de um banco de dados Oracle ou Real Application Cluster (RAC). Os dados armazenados em cache são distribuídos através de diversos nós ou IMDB Caches sem armazenamento compartilhado. Uma Grade de cache garante que os dados estão consistentes nos nós. Os membros da grade podem ser replicados.

A Figura 4 mostra uma Grade de cache composta por cinco membros de grade replicados. É possível adicionar membros de grade de forma incremental sem interrupção das operações. A configuração da replicação que precisa ser usada com o IMDB Cache é a configuração de par ativo-standby.

4.1 Definindo o conteúdo de um cache

O conteúdo de um Oracle IMDB Cache é definido através de uma sintaxe SQL estendida.

Um *Grupo de cache* é um subconjunto de tabelas do IMDB Cache que correspondem a um conjunto de tabelas do banco de dados Oracle usadas com frequência que estão relacionadas através de restrições de chaves externas. A sintaxe SQL é usada para definir os Grupos de cache e para escolher as colunas e linhas das tabelas do banco de dados Oracle que devem ser armazenadas em cache. Os usuários podem definir os Grupos de cache através de programação ou através de um utilitário interativo ttIsql.

Exemplo:

Considere que as seguintes tabelas existem em um banco de dados Oracle:

- Cliente (IdCliente, Nome, Idade, Sexo, Endereço, Estado, CEP, Telefone)
- Pedido (IdCliente, IdPedido, DataCompra, Quantidade)
- InteresseCliente (IdCliente, Interesse)

Um aplicativo pode querer armazenar em cache os perfis de clientes que fizeram pedidos desde 1º de janeiro de 2009. Para isso, ele pode definir os dois grupos de cache a seguir:

- O primeiro grupo de cache contém subconjuntos das três tabelas acima de clientes que fizeram pedidos desde 1º de janeiro de 2009 e que também vivem na região do Pacífico dos Estados Unidos. Além disso, o aplicativo pode escolher armazenar em cache somente um subconjunto de colunas das tabelas. Por exemplo, ele pode armazenar em cache as seguintes colunas:

- Cliente (IdCliente, Nome, Idade, Sexo, Estado)
- Pedido (IdCliente, DataCompra, Quantidade)
- InteresseCliente (IdCliente, Interesse)
- O segundo grupo de cache contém as mesmas informações que o primeiro grupo de cache, mas para os clientes nas regiões de montanhas dos Estados Unidos.

Os dois grupos de cache podem ser armazenados em diferentes nós executando o IMDB Cache.

Um conceito adicional usado pelo IMDB Cache é o de uma Instância do cache. Uma *Instância do cache* é uma coleção de registros relacionados que podem ser identificados de forma única e é usada para modelar um objeto complexo. Uma Instância do Cache forma a unidade do carregamento do cache e a duração do cache conforme descrito abaixo. No exemplo acima, todos os registros nas tabelas Cliente, Pedido e InteresseCliente que pertencem a um determinado ID de cliente (IdCliente) pertencem à mesma Instância do Cache e estão relacionadas umas às outras através de restrições de chaves externas. IdCliente identifica de forma exclusiva a Instância do Cache e é referida como a *Chave da Instância do cache*.

O TimesTen suporta os mesmos tipos de dados que o banco de dados Oracle.

Além de suportar seus próprios tipos de dados, o TimesTen suporta os mesmos tipos de dados básicos que o banco de dados Oracle de forma que não é necessário mapear os tipos de dados do banco de dados Oracle para os tipos de dados do TimesTen. Mas é possível mapear os tipos de dados do banco de dados Oracle para implementações mais eficientes do TimesTen. Por exemplo, um aplicativo pode mapear um tipo de dados NUMBER de um banco de dados Oracle para um tipo de dados INTEGER do TimesTen.

Observe que os desenvolvedores de aplicativos podem criar índices nas tabelas do cache in-memory. Os índices do cache in-memory podem corresponder aos índices no banco de dados Oracle ou podem ser diferentes. O designer de aplicativos pode usar a flexibilidade do TimesTen para criar múltiplos índices na mesma tabela e pode definir os índices em várias colunas.

4.2 Carregando dados e gerenciando o cache

Um aplicativo deve decidir como carregar os dados do Grupo de cache no IMDB Cache para processar. As seguintes técnicas estão disponíveis para o carregamento de dados:

- Carregamento explícito. Isso pode ser feito de diversas formas:

- Carregue o Grupo de cache inteiro de uma só vez. Isso é uma técnica adequada para usar se todo o conteúdo do Grupo de cache cabe no cache. A capacidade de descarregar um Grupo de cache inteiro também está disponível.
- Carregue as Instâncias do cache “por cláusula WHERE”. Neste caso, uma cláusula WHERE é usada para descrever o subconjunto das Instâncias do cache que devem ser trazidas para o cache. Os aplicativos também podem descarregar Instâncias do cache “por cláusula WHERE”.
- Carregue as Instâncias do cache “por ID”. Neste caso, uma lista de IDs de Instâncias do cache é usada para especificar as Instâncias do cache que devem ser trazidas para o cache. Os aplicativos também podem descarregar Instâncias do cache “por ID”.
- Carregamento dinâmico. Esta técnica está disponível para o carregamento de Instâncias do cache. O Carregamento Dinâmico é útil quando o Grupo de cache é muito grande para caber no cache e, portanto, somente o conjunto de trabalho de um aplicativo deve ser mantido no cache. Neste caso, os registros que compõem uma Instância do cache são carregados no cache automaticamente em uma falha do cache, por exemplo, quando uma declaração SQL² não encontra os dados solicitados no cache. Se a Instância do cache já está no cache, a declaração é manipulada diretamente do cache.

O Carregamento dinâmico está normalmente associado à *Duração do cache*. As Instâncias do cache podem expirar automaticamente no cache quando a capacidade do cache for excedida. O IMDB suporta *Duração com base na utilização* e *Duração com base no tempo*. A Duração com base na utilização usa um esquema LRU (usado pela última vez) para expirar as Instâncias do cache quando a capacidade do cache for excedida. A Duração com base no tempo concede às Instâncias do cache um *Tempo de vida* com uma certa duração no cache e exige a presença de uma coluna de carimbo de data e hora em uma das tabelas do Grupo de cache. O valor da coluna de carimbo de data e hora é gerenciado pelo aplicativo. As Instâncias do cache podem permanecer no cache desde que o valor de seu carimbo de data e hora mais o Tempo de vida não exceda a hora atual. Observe que a Duração do cache pode ser usada independentemente do carregamento dinâmico e pode na verdade ser usada com as tabelas normais do TimesTen de um banco de dados Oracle que não estão armazenadas em cache.

² O Carregamento dinâmico de uma Instância do cache está disponível para declarações SQL com uma expressão de igualdade na chave primária ou na chave externa para qualquer um dos registros na Instância do cache.

Um aplicativo pode escolher ter alguns Grupos de cache sujeitos à duração e outro não. Por exemplo, o aplicativo pode querer sempre manter as informações de catálogo no cache, mas pode querer carregar os perfis dos usuários sob demanda quando eles se conectarem ao aplicativo e expirar os perfis automaticamente quando os usuários desconectarem. As Instâncias do cache podem também ser explicitamente descarregadas pelo aplicativo.

Os dados carregados em tabelas do cache in-memory estão disponíveis para o processamento SQL, PL/SQL e Pro*C através de JDBC, ODBC, TTCclasses e OCI.

4.3 Compartilhando dados através de uma Grade de cache

Os Grupos de cache podem ser locais ou globais. Com *Grupos de cache locais*, os dados armazenados em cache não são compartilhados entre os membros da mesma Grade de cache. Os membros da grade podem ter dados desconexos ou sobrepostos, e o aplicativo deve determinar como os dados estão distribuídos entre eles. Por exemplo, os dados somente de leitura de catálogo podem ser armazenados em cache em todos os membros da grade para melhor desempenho e informações de clientes que são atualizadas podem ser particionadas por local geográfico em diferentes membros da grade. As atualizações confirmadas nas tabelas armazenadas em cache são propagadas para tabelas do banco de dados Oracle sem coordenação com outros membros da grade. Um Grupo de cache local pode ser definido como carregado explicitamente ou dinamicamente. Os Grupos de cache são locais por padrão, a menos que sejam definidos como globais.

Em um *Grupo de cache global*, os dados armazenados em cache são compartilhados entre os membros da mesma Grade de cache. O controle de simultaneidade é realizado através da grade e uma transação sendo executada em qualquer lugar na grade sempre vê a versão confirmada mais recente de uma Instância do cache. As atualizações confirmadas na mesma Instância da grade por diferentes membros da grade são propagadas para o banco de dados Oracle na ordem em que foram confirmadas dentro da grade para garantir a consistência dos dados.

4.4 Mantendo a consistência de dados

O Oracle IMDB Cache suporta atualizações aos dados armazenados em cache e mantém automaticamente a consistência entre os armazenamentos em cache e o banco de dados Oracle.

Os dados armazenados em cache podem ser atualizados no IMDB Cache ou no banco de dados Oracle. O IMDB Cache oferece o recurso de propagar automaticamente as atualizações do armazenamento em cache para o banco de dados Oracle e vice-versa. Entretanto, uma premissa adjacente é que o Grupo de cache seja praticamente todo atualizado ou exclusivamente atualizado no armazenamento em cache ou no banco de dados Oracle. É uma grande falha de projeto armazenar em cache um conjunto de tabelas que devem ser intensamente atualizadas no

cache e no banco de dados de back-end. Há, entretanto, casos em que é adequado permitir as atualizações em ambos. As atualizações no banco de dados Oracle podem, por exemplo, ocorrer somente à noite por questões de manutenção enquanto as atualizações ocorrem no(s) cache(s) durante o dia; ou as atualizações de dados centrais podem ocorrer no banco de dados Oracle, enquanto as atualizações de dados regionais ocorrem no(s) cache (caches).

Grupos de cache podem ser *gerenciados pelo sistema* ou *gerenciados pelo usuário*. Existem três tipos de Grupos de cache gerenciados pelo sistema:

- *Grupos de cache somente para leitura*. Esses Grupos de cache não podem ser atualizados no cache. Eles devem ser atualizados no banco de dados Oracle e o IMDB Cache gerencia a propagação das atualizações do banco de dados Oracle para o armazenamento em cache.
- *Grupos de cache AWT (Asynchronous Writethrough)*. Esses Grupos de cache podem ser atualizados no cache, mas não no banco de dados Oracle. O IMDB Cache propaga as atualizações do cache do banco de dados Oracle de forma assíncrona após a confirmação de uma transação.
- *Grupos de cache SWT (Synchronous Writethrough)*. Esses Grupos de cache podem ser atualizados no cache, mas não no banco de dados Oracle. As atualizações nas tabelas do cache in-memory são propagadas para o banco de dados Oracle de forma síncrona com a confirmação de uma atualização.

Os Grupos de cache gerenciados pelo sistema têm semânticas bem definidas e restrições para aplicar essas semânticas. Por outro lado, as semânticas dos Grupos de cache gerenciados pelo usuário são deixadas para o aplicativo. Por exemplo, um Grupo de cache gerenciado pelo usuário pode ser atualizado tanto no cache como banco de dados Oracle.

Grupos de cache somente para leitura, AWT, SWT e gerenciados pelo usuário devem ser todos Grupos de cache locais. Entretanto, somente Grupos de cache AWT dinâmicos podem ser especificados como Grupos de cache globais.

A tabela abaixo resume os diversos tipos de carregamento de Grupos de cache, compartilhamento de Grade de cache e opções de manutenção da consistência que estão disponíveis.

		Carregando dados em um Grupo de cache			
		Carregamento explícito		Carregamento dinâmico	
Mantendo a consistência de dados	Grupo de cache Somente para leitura	x		x	
	Grupo de cache AWT	x		x	x
	Grupo de cache SWT	x		x	
	Grupo de cache Gerenciado pelo usuário	x		x	
		Grupo de cache Local	Grupo de cache Global	Grupo de cache Local	Grupo de cache Global

Compartilhando dados através de uma Grade de cache

Os aplicativos do IMDB Cache podem enviar declarações SQL para um Grupo de cache ou para o banco de dados Oracle através de uma conexão única a um banco de dados do IMDB Cache. Esta capacidade de conexão única é habilitada através de um recurso *PassThrough* que verifica se a declaração SQL pode ser manipulada localmente pelas tabelas de cache in-memory ou se precisa ser direcionada para o banco de dados Oracle. O recurso *PassThrough* fornece as configurações que especificam quais tipos de declarações devem ser passadas e em quais circunstâncias. Uma configuração especialmente útil é aquela que especifica que todas as declarações que atualizam o banco de dados devem ser passadas para o banco de dados Oracle. Essa configuração permite que um aplicativo faça com que as atualizações sejam executadas no banco de dados Oracle e as leituras sejam executadas no IMDB Cache através de uma conexão única.

A seção abaixo descreve as operações do IMDB Cache disponíveis para manter a consistência dos dados armazenados. Algumas dessas operações são iniciadas automaticamente pelo IMDB Cache; outras são iniciadas explicitamente pelo aplicativo.

4.4.1 Propagação das atualizações do IMDB Cache para o banco de dados Oracle e entre membros da grade de cache para Grupos de cache globais

Como já observamos, os Grupos de cache globais também são Grupos de cache AWT dinâmicos. Um aplicativo com Grupos de cache globais será conectado a um dos membros da grade. Na maioria das vezes, ele acessará as Instâncias de cache que já estão armazenadas em cache no membro da grade. Entretanto, caso ele tente acessar uma Instância de cache que não esteja no membro da grade, o IMDB Cache irá carregar a Instância de cache dinamicamente de outro membro da grade ou do banco de dados Oracle, dependendo de onde reside a versão da Instância de cache mais recentemente atualizada. Isso é feito automaticamente, sem intervenção

do aplicativo. O IMDB Cache determina onde a cópia mais recente reside e usa comunicação "peer-to-peer" para trocar informações com outros bancos de dados IMDB Cache em sua grade.

Se uma transação atualiza uma Instância de cache em qualquer membro da grade, o seguinte mecanismo está disponível para manter o banco de dados Oracle sincronizado com o cache:

- *Propagar.* O IMDB Cache propaga as atualizações para o banco de dados Oracle após a transação ser confirmada. Se outra transação atualiza a mesma Instância de cache em outro membro da grade pouco tempo depois, o IMDB Cache garante que as confirmações sejam propagadas para o banco de dados Oracle na ordem correta.

A Figura 5 mostra uma Grade de cache composta por três membros de grade. Todos os membros de grade armazenam o mesmo Grupo de cache global e cada membro de grade tem em seu cache somente algumas Instâncias de cache do Grupo de cache global. Essas instâncias são armazenadas em cache porque foram recentemente acessadas em seus respectivos membros de grade. Com o passar do tempo, cada instância continua a ser acessada em seu membro de grade e, portanto, permanece lá, ou é acessada em outro membro de grade e é movida para esse membro, ou não é acessada por nenhum membro e, neste caso, expira totalmente da Grade de cache.

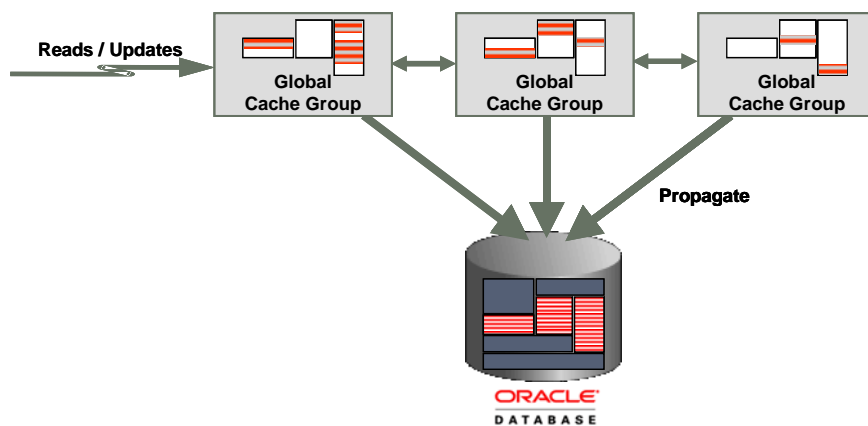


Figura 5. Propagação de atualizações e consistência de cache para Grupos de cache globais

4.4.2 Propagação de atualizações do IMDB Cache para um banco de dados Oracle para Grupos de cache locais

Para Grupos de cache locais que podem estar atualizados no cache, os seguintes mecanismos estão disponíveis para manter o banco de dados Oracle sincronizado com o cache:

- *Propagar.* Com a opção de propagação ativada, todas as modificações em um Grupo de cache, ou seja, todas as operações de insert, update e delete são automaticamente propagadas

para o banco de dados Oracle. O momento em que a propagação ocorre é diferente para grupos de cache SWT e AWT. Com Grupos de cache SWT, quando o aplicativo conclui uma transação que tenha modificado um ou mais Grupos de cache, a transação é primeiro confirmada no banco de dados Oracle e, em seguida, no IMDB Cache. Essa técnica permite que o banco de dados Oracle aplique qualquer lógica necessária relacionada aos dados antes de eles serem confirmados no IMDB Cache. Com os Grupos de cache AWT, quando o aplicativo conclui uma transação, a transação é confirmada no IMDB Cache e o controle volta para o aplicativo. As alterações feitas são então propagadas de forma assíncrona para o banco de dados Oracle.

- *Liberar*. Essa operação é direcionada por uma solicitação explícita do aplicativo e pode ser aplicada aos Grupos de cache ou Instâncias de cache. Ela só é permitida em Grupos de cache ou Instâncias de cache que tenham a opção Propagar desligada. A operação atualiza os registros no banco de dados Oracle com os valores dos registros no cache. Essa operação é útil quando atualizações frequentes acontecem por determinado período de tempo envolvendo o mesmo conjunto de registros. Em vez de propagar um "play-by-play" de cada atualização, a imagem final de cada registro é enviada ao banco de dados Oracle e aplicada nele.

Uma aplicação pode configurar uma Grade de cache com diversos Grupos de cache locais, que podem ser atualizados, em diferentes membros da grade. A propagação de atualizações dos membros da grade para o banco de dados Oracle será gerenciada pelo IMDB Cache, mas é recomendado que os Grupos de cache locais em diferentes membros da grade não sejam sobrepostos para que atualizações diferentes dos mesmos dados não ocorram em nós diferentes, resultando em valores de dados imprevisíveis no back-end.

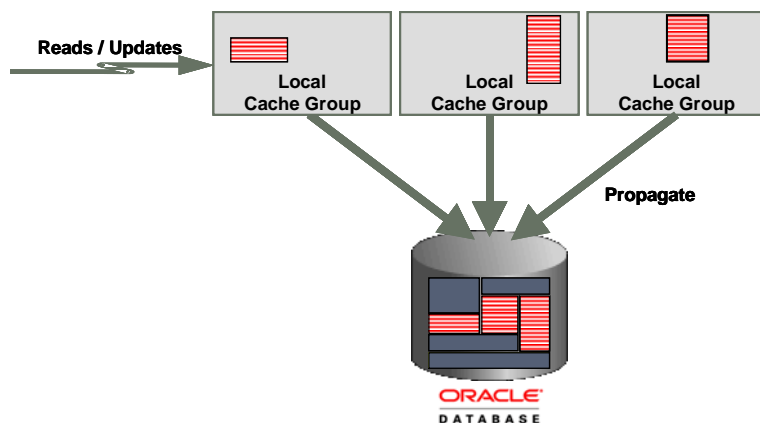


Figura 6. Propagação de atualizações para Grupos de cache locais que podem ser atualizados

4.4.3 Propagação de atualizações do banco de dados Oracle para o IMDB Cache para Grupos de cache locais

Para um Grupo de cache local³ que é atualizado no banco de dados Oracle, os seguintes mecanismos estão disponíveis para manter o conteúdo do cache sincronizado com o banco de dados Oracle:

- *Atualizar*. Esta é uma solicitação explícita do aplicativo para atualizar um Grupo de cache inteiro ou Instâncias de cache específicas. É equivalente a uma operação de descarga seguida de uma operação carga.
- *Autoatualização total*. Com a Autoatualização total, o aplicativo indica com que frequência as atualizações devem ocorrer e o IMDB Cache atualiza automaticamente o Grupo de cache nos intervalos de tempo indicados pelo aplicativo.
- *Autoatualização incremental*. Ao contrário da Autoatualização total, uma Autoatualização incremental atualiza somente os registros que foram modificados no banco de dados Oracle desde a última atualização. Assim como ocorre com a Autoatualização total, o aplicativo deve indicar a frequência das atualizações e o IMDB Cache realiza automaticamente a atualização incremental na frequência especificada.

³ Observe que os Grupos de cache globais podem não ser atualizados no banco de dados Oracle.

A Autoatualização incremental pode ser usada com Duração com base no tempo para manter uma "janela deslizante" no cache. Por exemplo, um aplicativo de atendimento ao cliente pode desejar manter no cache todos os incidentes relatados nos últimos 5 dias. Neste caso, ele pode especificar que o Grupo de cache deve usar a Autoatualização incremental e a Duração com base no tempo deve ter um tempo de vida de 5 dias. Conforme novos incidentes são inseridos no banco de dados Oracle, a Autoatualização incremental propaga os incidentes automaticamente para as tabelas de cache in-memory. Se esses incidentes forem atualizados no banco de dados Oracle, as atualizações serão propagadas automaticamente para as tabelas de cache in-memory. Os incidentes devem conter um carimbo de data e hora mantido pelo aplicativo. Uma vez que o valor do carimbo de data e hora é mais que 5 dias mais antigo que a data atual, o incidente associado irá expirar do cache automaticamente.

As três técnicas descritas acima são úteis em diferentes circunstâncias. Considere que um Grupo de cache precisa ser atualizado somente uma vez por dia às 2:00 quando as atividades em um site provedor de conteúdo forem mínimas. Neste caso, a Autoatualização total pode ser a melhor escolha. Por outro lado, um Grupo de cache que precisa ser atualizado uma vez a cada cinco minutos deve usar a Autoatualização incremental. Finalmente, um Grupo de cache que precisa ser atualizado raramente, mas em horários imprevisíveis são conhecidos apenas pelo aplicativo, deve usar a opção Atualizar.

Uma aplicação pode configurar uma Grade de cache com diversos Grupos de cache locais somente para leitura em diferentes membros da grade. Os Grupos de cache nos diferentes membros da grade podem estar totalmente desconexos, parcialmente sobrepostos ou serem idênticos. A propagação das atualizações do banco de dados Oracle para todos os membros da grade será gerenciada pelo IMDB Cache.

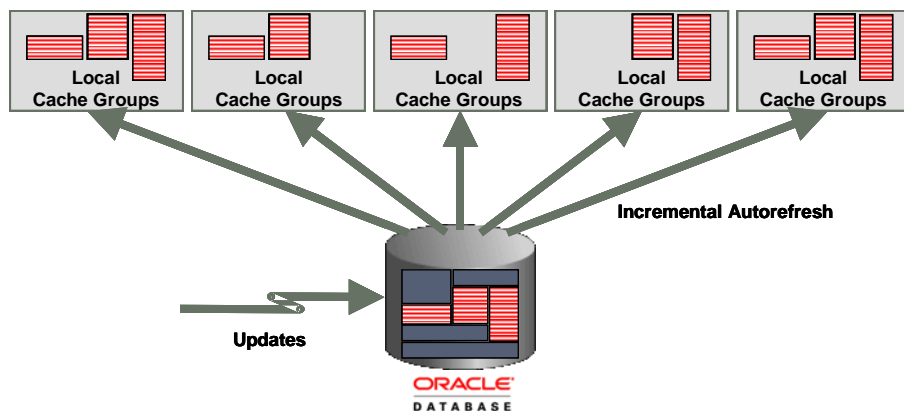


Figura 7. Autoatualização incremental de Grupos de cache locais somente para leitura

4.5 Alta disponibilidade

O IMDB Cache suporta alta disponibilidade através da camada de aplicativos e da camada do servidor do banco de dados.

Quando o Oracle TimesTen é usado exclusivamente como o banco de dados de registro em vez de um cache de banco de dados in-memory do banco de dados Oracle, ele garante a alta disponibilidade de seus dados através da replicação, diversas operações on-line e inúmeros utilitários que suportam falhas, recuperação e atualizações on-line. A detecção automática de falhas e o failover do banco de dados e aplicativos estão disponíveis através da integração com o Oracle Clusterware. Da mesma forma, o banco de dados Oracle suporta alta disponibilidade de seus dados através de um conjunto de recursos que incluem o Oracle Real Application Clusters (RAC), Oracle Automatic Storage Management (ASM) e o Oracle Data Guard. Além disso, o componente de replicação do IMDB Cache fornece vários recursos que garantem a alta disponibilidade dos dados armazenados em cache, e a recuperação automática de falhas que se espalham pela camada de aplicativos e pelo banco de dados Oracle na camada de banco de dados. Eles estão descritos abaixo.

4.5.1 Manipulação de falha de um nó do cache In-Memory

Para proteção contra falhas dos nós do cache e garantia de disponibilidade contínua dos dados armazenados, a replicação de dados do TimesTen proporciona a manipulação de falhas e recuperação dos nós de cache. A configuração de replicação de Par Ativo-Standby com múltiplos assinantes somente para leitura é projetada para incluir um banco de dados Oracle como parte da configuração e proporcionar o failover e a recuperação dos nós do cache.

Com o Par Ativo-Standby, todas as atualizações são sempre aplicadas em primeiro lugar no nó ativo. As atualizações são então replicadas no nó em standby e, em seguida, replicadas do nó em standby para todos os nós de Assinantes somente para leitura. Desta forma, o nó em standby está sempre à frente de todos os nós de Assinantes somente para leitura e, portanto, se um nó ativo para de funcionar, não há dúvidas sobre qual dos nós de assinantes deve se tornar o novo nó ativo.

Grupos de cache somente para leitura

A configuração de replicação de Par Ativo-Standby é projetada para funcionar tanto com Grupos de cache somente para leitura como Writethrough. Com os Grupos de cache somente para leitura, as atualizações que são aplicadas no banco de dados Oracle são propagadas para o nó ativo somente. A replicação do TimesTen propaga então as atualizações para todos os outros nós indo através do nó em Standby primeiro.

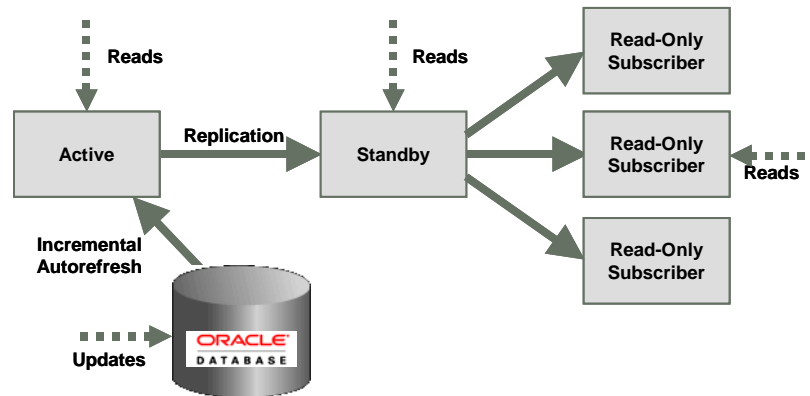


Figura 8. Grupos de cache somente para leitura usando a configuração de Par Ativo-Standby

Se o nó Ativo falhar, o nó em Standby se torna o novo nó Ativo. A partir desse momento, as atualizações do banco de dados Oracle são propagadas para o novo nó Ativo (o antigo nó em Standby) e são, em seguida, replicadas para os Assinantes somente para leitura. Uma vez que o antigo nó ativo fica novamente on-line, ele se torna o novo nó em Standby. A replicação do TimesTen manipula automaticamente a troca da propagação de atualizações do nó Ativo para o nó em Standby e a recuperação do nó com problema.

Da mesma forma, se o nó em Standby falhar, a replicação do nó Ativo será automaticamente redirecionada para os nós dos Assinantes somente para leitura. Uma vez que o nó em Standby volta a ficar on-line, a replicação irá garantir que ele tome conhecimento de todas as atualizações perdidas antes de retomar seu papel como nó em Standby.

Grupos de cache Writethrough

Com os Grupos de cache Writethrough, as atualizações são aplicadas no nó ativo. Em seguida, elas são replicadas para o nó em Standby. Uma vez ali, elas são propagadas para o banco de dados Oracle e replicadas para todos os nós de Assinantes somente para leitura.

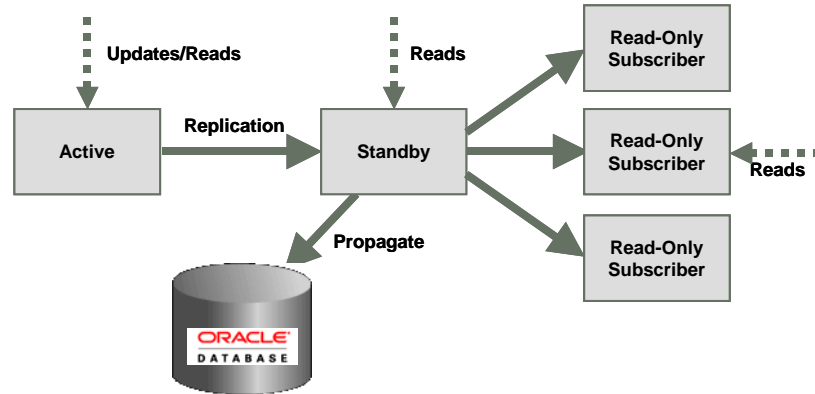


Figura 9. Grupos de cache Writethrough usando a configuração Ativo-Standby

Se o nó Ativo falhar, o nó em Standby se torna o novo nó Ativo. A partir deste momento, todas as atualizações devem ser enviadas para o novo nó Ativo, ou seja, o antigo nó em Standby. As atualizações no novo nó Ativo serão propagadas para o banco de dados Oracle e replicadas para os nós de Assinantes somente para leitura. Uma vez que o antigo nó ativo fica novamente on-line, ele se torna o novo nó em Standby. A replicação do TimesTen manipula automaticamente a recuperação do novo nó em Standby e a transferência da responsabilidade de propagar as atualizações no backend para o novo nó em Standby.

Da mesma forma, se o nó em Standby falhar, a replicação do nó Ativo será automaticamente redirecionada para os nós de Assinantes somente para leitura e o nó Ativo começará a propagar atualizações diretamente para o banco de dados Oracle. Uma vez que o nó em Standby fica novamente on-line, a replicação garantirá que ele tome conhecimento de todas as atualizações que perdeu antes de assumir novamente seu papel como nó em Standby e fará com que o nó em Standby volte a propagar atualizações para o banco de dados Oracle e os Assinantes somente para leitura.

4.5.2 Manipulação de falhas no banco de dados Oracle

Se o banco de dados Oracle fica inacessível para o IMDB Cache por qualquer motivo como uma falha na rede, falha de hardware ou falha no banco de dados Oracle, o IMDB Cache é projetado para ser flexível a essas falhas. O cache in-memory continuará a estar acessível para os aplicativos. Além disso, no caso de um Grupo de cache AWT, as atualizações no cache continuarão a ser registradas no Oracle TimesTen de forma que, uma vez que o banco de dados Oracle fique novamente acessível, as atualizações sejam propagadas para ele. Da mesma forma, alterações em Grupos de cache somente para leitura feitas no banco de dados Oracle mas ainda não propagadas para o cache in-memory permanecerão gravadas no banco de dados Oracle e

serão propagadas para o(s) cache(s) uma vez que o banco de dados Oracle estiver acessível novamente.

Além disso, o IMDB Cache tira proveito integral dos recursos de alta disponibilidade do RAC. Uma configuração RAC consiste de um único banco de dados físico acessível por diversos nós. A configuração em tempo de execução em um único nó é chamada *Instância*. O RAC proporciona o equilíbrio de carga, alta disponibilidade e consistência de dados entre todas as instâncias.

O IMDB Cache recupera rapidamente de uma falha de nó RAC sem exigir a intervenção do usuário. Para fazer isso, o IMDB Cache usa os recursos Transparent Application Failover (TAF) e Fast Application Notification (FAN) da Oracle sempre que disponíveis, ou seja, dependendo da versão do cliente Oracle, servidor e a configuração TAF. Se uma instância Oracle à qual o IMDB Cache está conectado falhar, a conexão será automaticamente direcionada para outra instância. Se uma operação de Atualização, Autoatualização total ou Autoatualização incremental estiver em andamento quando a falha ocorrer, a operação irá desfazer automaticamente as alterações que ocorreram no banco de dados in-memory e reiniciará a operação. Se uma operação de Propagação para um Grupo de cache AWT estiver em andamento quando uma falha ocorrer, a transação irá desfazer automaticamente as alterações que ocorreram no banco de dados Oracle caso precisem ser desfeitas e reiniciará a operação de Propagação.

Se o banco de dados Oracle for replicado para um banco de dados em standby usando Data Guard síncrono, no caso de uma falha do banco de dados Oracle ativo, o IMDB Cache fará o failover automaticamente para o banco de dados Oracle em standby sem perda de dados.

5. Desempenho

Para medir o desempenho do IMDB Cache, nós desenvolvemos uma referência que simula um aplicativo HLR (Home Location Register) como aqueles usados em redes celulares. A referência consiste de um conjunto de 7 transações, cada uma modelando uma operação típica executada por um HLR como configurar ou excluir o encaminhamento de chamadas ou atualizar informações sobre um assinante de telefone celular.

Executamos a referência em duas configurações diferentes. Na primeira configuração, o aplicativo HLR foi executado com um banco de dados Oracle, com o aplicativo de referência sendo executado em um servidor e o Oracle Database 10g em outro servidor. Na segunda configuração, adicionamos o IMDB Cache à frente do banco de dados Oracle; o programa do aplicativo HLR foi vinculado diretamente ao banco de dados de cache do TimesTen sendo executado em um servidor e o Oracle Database 10g sendo executado no outro servidor. Os dados em cache foram armazenados em Grupos de cache AWT, permitindo que todas as

atualizações nos dados armazenados em cache fossem propagadas automaticamente para o banco de dados Oracle.

O aplicativo foi implementado em Java usando JDBC para acesso dos dados. Todos os quatro servidores tinham configurações idênticas com 6 GB de RAM física, dois processadores Intel Xeon 2.4 GHz com hyper-threading sendo executados no Oracle Enterprise Linux 5.2.

Medimos o tempo de resposta médio para cada tipo de transação ao executar no banco de dados Oracle e no IMDB Cache. O gráfico abaixo mostra uma redução significativa do tempo de resposta do aplicativo ao usar o IMDB Cache.

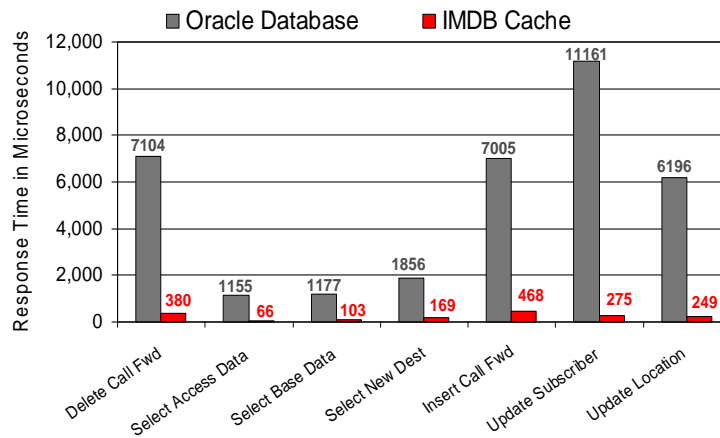


Figura 10. Comparação do tempo de resposta para o aplicativo HLR de referência

Também medimos o throughput combinado de todas as transações nas duas configurações. O gráfico abaixo mostra um aumento significativo no throughput do aplicativo ao usar o IMDB Cache.

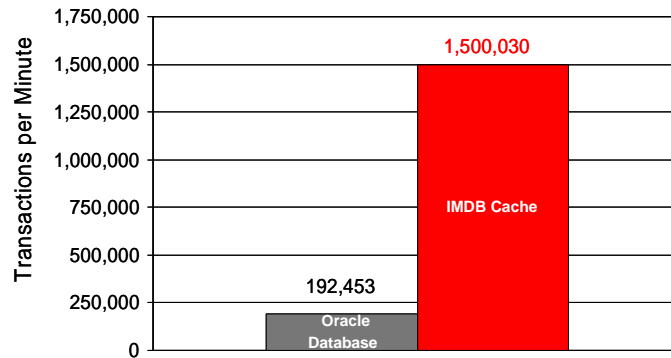


Figura 11. Comparação de throughput para o aplicativo HLR de referência

Esta referência mostra os benefícios da utilização do IMDB Cache. Conforme mostrado nos gráficos acima, o tempo de resposta do aplicativo aumentou em um fator de 10 a 40 e o throughput total melhorou mais de 7 vezes. Em geral, a proporção da melhoria com o IMDB Cache varia de acordo com o hardware e as plataformas.

6. Exemplos

Nesta seção, examinamos alguns cenários de armazenamento em cache e as configurações do IMDB Cache recomendadas e os tipos de Grupo de cache para esses cenários. Observe que, apesar de os exemplos se concentrarem em um tipo de Grupo de cache específico, tipos diferentes de Grupos de cache podem coexistir no mesmo IMDB Cache para melhor se adequarem às necessidades de uma aplicação.

6.1 Cache somente para leitura

Os Grupos de cache somente para leitura com Autoatualização incremental são ideais para o armazenamento em cache de dados frequentemente referenciados.

Existem muitas aplicações que podem se beneficiar de caches somente para leitura. As principais características dessas aplicações são que alguns conjuntos de registros são consultados continuamente. Esses registros podem ser atualizados com frequência ou não, mas a proporção de leituras em relação a gravações é alta. Exemplos desses tipos de registros incluem listas de preços como aplicativos de comércio eletrônico, tabela de horários de companhias aéreas para aplicativos de reservas aéreas e disponibilidade de quartos em aplicativos de reserva em hotéis.

A melhor configuração de cache para esses dados é um Grupo local somente para leitura (gerenciado pelo sistema) com Autoatualização incremental. Os dados devem ser atualizados no

banco de dados de back-end. As atualizações serão propagadas automaticamente para o cache. A frequência da propagação é determinada pelo aplicativo e deve depender da frequência de atualizações no back-end e a moeda dos dados exigidos pelo aplicativo.

Se o cache deve ser implantado em múltiplos membros de grade de uma Grade de cache, é necessário definir os Grupos de cache locais nos membros de grade e cada membro será atualizado diretamente a partir do banco de dados de back-end.

Observe que um aplicativo de comércio eletrônico não precisará normalmente atualizar as listas de preços armazenadas em cache com frequência uma que alterações em listas de preços são raras. Por outro lado, um aplicativo de rastreamento de voos, ao mesmo tempo em que precisa ser lido intensamente, também precisa manter o status dos voos razoavelmente atualizados no cache. As atualizações são mais bem aplicadas no banco de dados Oracle com um intervalo de Autoatualização incremental razoavelmente pequeno, por exemplo, 5 minutos, definido para Grupos de cache somente para leitura. O IMDB Cache irá propagar automaticamente todas as atualizações para os membros da grade que incluem os dados atualizados. Um aplicativo conectado a um membro da grade pode configurar uma conexão única com o IMDB Cache e usar a opção PassThrough para rotear todas as atualizações para o banco de dados Oracle enquanto executa todas as leituras no cache.

6.2 Cache de "janela deslizante" somente para leitura

Grupos de cache somente para leitura com Autoatualização incremental e Duração com base em tempo são ideais para armazenar em cache dados referenciados com frequência que caem em uma "janela deslizante".

Em muitos casos, os dados somente para leitura exigidos por um aplicativo são dados com um componente de tempo onde dados mais novos são acessados com mais frequência que dados mais antigos. Dados novos são gerados constantemente e os dados mais antigos acabam tendo menos valor para essa classe específica de aplicativos. Portanto, alguém pode pensar em um intervalo de tempo de tamanho fixo que avança constantemente com dados chegando no intervalo em uma ponta e saindo do intervalo na outra ponta. O aplicativo está interessado somente nos dados que estão dentro do intervalo, normalmente chamado de uma "janela deslizante".

Exemplos de aplicativos que podem precisar de dados que caem em uma "janela deslizante" são aplicativos do mercado de ações que podem desejar obter os 3 últimos dias de negociações ou um aplicativo de fornecimento de notícias que pode desejar obter as últimas 24 horas de novos vídeos.

Para armazenar em cache os dados que caem em uma "janela deslizante", queremos que os dados novos sejam trazidos para o cache automaticamente e que os dados antigos expirem do cache automaticamente. Também queremos que os dados que estejam no cache sejam atualizados

automaticamente quando forem feitas alterações no back-end. A melhor configuração de cache para esses dados é um Grupo de cache somente para leitura (gerenciado pelo sistema) com Autoatualização incremental e duração com base no tempo.

Como no exemplo anterior, se o cache deve ser implantado em múltiplos membros de grade de uma Grade de cache, é necessário definir os Grupos de cache locais nos membros de grade e cada membro será atualizado diretamente a partir do banco de dados Oracle.

6.3 Cache que pode ser atualizado

Os Grupos AWT são ideais para caches que podem ser atualizados.

Alguns aplicativos precisam de atualizações imediatas em tempo real dos dados armazenados em cache com eventual propagação de atualizações para um banco de dados Oracle. Por exemplo, um aplicativo que gerencia e fornece serviços de assinatura de telefone e autentica o acesso desses serviços irá normalmente armazenar informações do assinante no IMDB Cache. As alterações no serviço de um usuário devem ser refletidas imediatamente no cache e devem ser propagadas para o banco de dados de back-end.

A melhor configuração para esses dados é um Grupo de cache AWT (Asynchronous Writethrough).

Se o número de assinantes é grande o suficiente para exigir que o aplicativo seja implantado em múltiplos membros de grade de uma Grade de cache, os Grupos de cache locais devem ser definidos nos membros de grade com o Grupo de cache em cada membro possuindo um subconjunto do total de assinantes e sem sobreposição dos subconjuntos de diferentes membros da grade. Por exemplo, os assinantes podem ser particionados por código de área.

6.4 Cache dinâmico que pode ser atualizado

Os Grupos de cache globais AWT com Carregamento dinâmico e Duração com base na utilização são ideais para os caches dinâmicos que podem ser atualizados.

Para alguns aplicativos, o acesso aos dados ativos deve ser muito rápido, mas o conjunto de dados ativos varia com o tempo e é um subconjunto de uma quantidade muito maior de dados que não pode ser mantida por inteiro em um cache por ser muito grande. Os dados ativos precisam ser trazidos para o cache sob demanda e o conteúdo do cache precisa ser dinâmico de forma que os dados ativos possam substituir os dados obsoletos.

Um exemplo desse tipo de aplicação é um call center que gerencia um grande volume de sessões de atendimento simultâneas. O aplicativo será normalmente implantado em diversos nós do servidor de aplicativos. Os clientes que entram em contato com o call center são

automaticamente direcionados para um nó disponível do servidor de aplicativos e o perfil do cliente correto deve idealmente estar disponível neste nó do servidor.

A melhor configuração neste caso é um Grupo de cache global AWT com carregamento dinâmico e duração com base na utilização e com um membro de grade configurado em cada nó do servidor de aplicativos.

Com esta configuração, quando um cliente for direcionado para um nó disponível do servidor, o perfil do cliente correto será carregado dinamicamente a partir do banco de dados Oracle para o membro da grade. Quando o cliente concluir uma chamada, as alterações em seu perfil serão propagadas do IMDB Cache para o banco de dados Oracle. A duração LRU (usado pela última vez) removerá automaticamente os perfis de clientes inativos do IMDB Cache. Se o mesmo cliente entrar em contato com o call center pouco tempo após a primeira chamada e for direcionado para um nó diferente, o perfil do cliente será carregado dinamicamente no novo nó do banco de dados Oracle ou do membro da grade onde ele havia sido anteriormente carregado dependendo de onde estiver a cópia mais recente. O IMDB Cache determina onde está a cópia mais recente. Ele também gerencia as atualizações simultâneas dos dados na grade.

Todos os dados do cliente são armazenados no banco de dados Oracle. O banco de dados Oracle é muito maior que os bancos de dados do IMDB Cache combinados e pode ser acessado de forma mais eficiente por aplicativos que não exigem o desempenho em tempo real do IMDB Cache mas que exigem o acesso a uma grande quantidade de dados. Esses aplicativos podem incluir um aplicativo de cobrança e um aplicativo de data mining.

Conforme a base de clientes e a demanda para atender mais clientes simultaneamente aumentam, o call center pode decidir implantar nós adicionais do servidor de aplicativos. Os novos membros do IMDB Cache podem ser unidos à grade do IMDB Cache sem interrupção das solicitações em andamento na grade. Da mesma forma, falhas ou a remoção de nós individuais não interrompem as operações no resto da grade.

6.5 Cache de captura de dados com Taxa de chegada desigual

Os Grupos de cache AWT com Duração com base na utilização são ideais para capturar dados com taxas de chegada desiguais.

Existe uma classe de aplicativos onde os novos dados são gerados a uma taxa muito alta por um determinado período de tempo e a uma taxa moderada em outros períodos. Durante os períodos de alta atividade, o banco de dados de back-end fica frequentemente impossibilitado de acompanhar o alto throughput exigido pela aplicação. Essas aplicações podem se beneficiar de um cache que, na verdade, acaba “suavizando” a taxa de chegada dos dados recém-gerados.

Por exemplo, um aplicativo que informa os valores das ações terá uma taxa de chegada de novos valores que varia muito com o tempo. Ela será particularmente alta quando o mercado abrir e fechar, e será menor nos outros períodos. O pico da taxa de chegada de dados não pode normalmente ser manipulado por um banco de dados com base em disco, mas pode ser mantido pelo IMDB Cache.

A melhor configuração de cache para esses dados é um Grupo de cache AWT (gerenciado pelo sistema) com Duração com base na utilização. As inserções nos Grupos de cache AWT são automaticamente propagadas para o banco de dados Oracle de back-end. E a Duração com base na utilização removerá automaticamente os dados do cache in-memory para liberar espaço.

6.6 Cache de captura de dados com Taxa de chegada alta constante

Os Grupos de cache AWT com duração com base na utilização associados a tabelas normais do TimesTen com duração com base na utilização são ideais para capturar dados com uma taxa de chegada constantemente alta.

Há outra classe de aplicações onde os novos dados são gerados a uma taxa alta, mas onde a alta taxa de chegada não necessariamente diminui. Armazenar em cache dados temporários com a taxa de chegada muito alta para ser absorvida por um banco de dados de back-end não resolve o problema se a taxa de chegada não diminui, pois não há um período de descanso para o banco de dados de back-end se atualizar. Entretanto, para essas aplicações, quase sempre os dados recém-gerados podem ser agregados em uma forma mais condensada antes de serem permanentemente armazenados no banco de dados de back-end. Também é comum para essas aplicações analisar os dados coletados em tempo real para detectar padrões interessantes ou irregulares.

Um exemplo desse tipo de aplicação é o que coleta dados de sensores ou leitores de RFID. Os dados são quase sempre repetitivos e podem ser facilmente agregados e frequentemente precisam de análise em tempo real.

A melhor configuração para essas aplicações é inserir os dados conforme eles chegam a uma ou mais tabelas gerenciadas somente pelo Oracle TimesTen, ou seja, não há imagem desses dados no banco de dados de back-end. As tabelas não armazenadas em cache, presentes somente no TimesTen, podem ser configuradas com Duração com base na utilização. Uma vez que os dados são agregados pelo aplicativo, eles podem ser inseridos no cache em um Grupo de cache AWT (gerenciado pelo sistema) com Duração com base na utilização. O IMDB Cache irá propagar automaticamente todos os agregados para o banco de dados de back-end. Uma vez que tanto as tabelas somente no TimesTen como as tabelas no cache estão configuradas com a Duração com base na utilização, os registros usados menos recentes irão expirar automaticamente para abrir espaço no cache para novos registros.

6.7 Cache gerenciado pelo usuário que pode ser atualizado

Os Grupos de cache gerenciados pelo usuário com Liberação explícita são os mais indicados para aplicações com atualizações frequentes, mas raras transações de negócios.

Algumas aplicações precisam executar diversas atualizações no cache para obter melhor desempenho, mas precisam registrar permanentemente a transação final no banco de dados Oracle. Um exemplo deste tipo de aplicação é um aplicativo de comércio eletrônico onde ele deve manter um número de carrinhos de compra para os usuários ativos. Os carrinhos de compra serão atualizados repetidamente no cache. Essas atualizações não precisam ser propagadas para o banco de dados Oracle uma vez que são de pouco valor. Entretanto, uma vez que um usuário executa uma compra, a transação precisa ser gravada permanentemente no banco de dados Oracle.

A melhor configuração para esse tipo de dados é um Grupo de cache que pode ser atualizado (gerenciado pelo usuário) onde o aplicativo envia solicitações de Liberação explícitas sempre que é necessário gravar uma transação no banco de dados Oracle. Essa configuração pode ser associada com a Duração com base na utilização de forma que os carrinhos de compra abandonados possam ser excluídos do cache automaticamente.

6.8 Cache distribuído dinâmico somente para leitura

As tabelas somente para leitura com carregamento dinâmico e Duração com base na utilização são mais indicadas para um cache distribuído dinâmico somente leitura.

Em alguns casos, uma aplicação pode ser distribuída por muitos nós para manipular uma taxa de throughput que não pode ser manipulada por um único nó, e o conjunto de dados ativos exigido pela aplicação é dinâmico e é, em qualquer momento, um subconjunto muito menor de todo o conjunto de dados. Um exemplo desse tipo de aplicação pode ser um aplicativo de comercialização onde os dados ativos são os perfis dos negociantes ativos.

A melhor configuração para esses dados é configurar o cache in-memory em cada nó com um grupo de cache somente para leitura sobre o mesmo conjunto de tabelas no banco de dados Oracle e usar o Carregamento dinâmico e a Duração com base na utilização com essas tabelas de cache. O que irá acontecer é que cada nó terá os perfis dos negociantes que ele precisa que sejam carregados automaticamente conforme necessário e os perfis irão expirar dos caches quando não forem mais necessários para liberar espaço para os perfis necessários.

7. Conclusão

O Oracle In-Memory Database Cache permite que você melhore o tempo de resposta das transações da aplicação armazenando em cache um subconjunto de tabelas e fragmentos de

tabelas de desempenho crítico de um banco de dados Oracle para a camada de aplicativos. Ao contrário do que acontece com mecanismos de cache de resultados simples, os aplicativos podem executar comandos SQL e PL/SQL nos dados armazenados em cache uma vez que as tabelas armazenadas em cache são gerenciadas como tabelas de um banco de dados relacional no TimesTen In-Memory Database. Os armazenamentos em cache podem ser compartilhados entre diferentes aplicativos. As atualizações podem ser aplicadas nos armazenamentos em cache e eles são mantidos consistentes com o banco de dados Oracle. O armazenamento de dados em cache através do IMDB Cache é superior às outras técnicas de armazenamento em cache, pois oferece total funcionalidade relacional, escalabilidade incremental associada à transparência local, excelente desempenho, manutenção automática da consistência de dados com o banco de dados Oracle e alta disponibilidade entre camadas para os aplicativos sendo executados na camada de aplicativos

Ao trazer os dados para mais perto do aplicativo e ao processar as consultas em um banco de dados in-memory, o Oracle In-Memory Database Cache reduz significativamente o tempo de resposta. Ao tirar a sobrecarga de parte do processamento de dados do servidor do banco de dados Oracle, o throughput geral da aplicação é melhorado significativamente sem interferir no gerenciamento e administração centralizados do banco de dados de back-end.

8. Referências

1. Desempenho extremo através do Oracle TimesTen In-Memory Database. *Um artigo técnico da Oracle, julho de 2009.*



Usando o Oracle In-Memory Database Cache
para acelerar o banco de dados Oracle
Julho de 2009

Oracle do Brasil Sistemas Ltda.
Sede no Brasil
Av. Alfredo Egydio de Souza Aranha, 100
São Paulo, SP
Brasil

CNPJ: 59.456.277/0001-76
Fone: (0xx11) 5189-1000
oracle.com



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

Copyright © 2008, 2009, Oracle e/ou suas afiliadas. Todos os direitos reservados. Este documento é fornecido apenas para fins informativos e seu conteúdo está sujeito a alteração sem aviso prévio. Não há garantias de que este documento esteja isento de erros e nem que esteja sujeito a outras garantias ou condições legais, expressas ou implícitas, incluindo garantias ou condições de comercialização e uso para um propósito específico. A Oracle isenta-se de qualquer responsabilidade em relação a este documento, sendo que ele não representa qualquer obrigação contratual direta ou indireta. Este documento não pode ser reproduzido ou transmitido de qualquer forma ou através de qualquer meio, seja eletrônico ou mecânico, para qualquer objetivo, sem a permissão expressa por escrito da Oracle.

Oracle é uma marca comercial registrada da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.