

Oracle TimesTen – cała baza danych w pamięci operacyjnej

Mikołaj Morzy
Instytut Informatyki Politechniki Poznańskiej

Mikolaj.Morzy@put.poznan.pl

Abstrakt. Oracle TimesTen to rodzina technologii przetwarzania danych w pamięci ulotnej. Główną cechą technologii TimesTen jest przeniesienie przetwarzania danych do warstwy aplikacyjnej i całkowita rezygnacja z wykorzystania pamięci dyskowej. Struktury danych, indeksy, czy algorytmy optymalizacji zapytań są specjalnie zaprojektowane zgodnie z założeniem, że wszystkie wymagane dane znajdują się w pamięci ulotnej o dostępie swobodnym. Taka architektura zapewnia minimalne czasy odpowiedzi, niespotykaną wcześniej przepustowość, oraz możliwość zarządzania transakcjami i zdarzeniami w bazach danych składających się z ogromnych wolumenów danych. Z drugiej strony, architektura bazy danych w pamięci operacyjnej wymusza opracowanie nowych mechanizmów zachowania spójności i odtwarzania po awarii.

Technologia TimesTen jest wykorzystywana w dwóch produktach: Oracle TimesTen In-Memory Database oraz Oracle In-Memory Database Cache. Pierwszy produkt to samodzielny system zarządzania bazą danych, wykorzystujący przechowywanie danych w pamięci ulotnej i okresową nieblokującą synchronizację bazy danych z obrazem dyskowym oraz replikację danych do innych systemów. Drugi produkt umożliwia wykorzystanie bazy danych TimesTen jako pamięci podręcznej dla tradycyjnej relacyjnej bazy danych. W takim przypadku baza danych TimesTen zawiera kopię wybranych krytycznych podzbiorów danych i oferuje płynną synchronizację danych między pamięcią podręczną przechowywaną w bazie TimesTen a relacyjną bazą danych, zapewniając jednocześnie bardzo efektywny dostęp do danych przechowywanych w tak skonfigurowanej pamięci podręcznej. W niniejszym artykule zaprezentowano ogólną architekturę bazy danych TimesTen, przedstawiono charakterystyczne elementy bazy danych TimesTen (pliki kontrolne, indeksy, dziennik bazy danych) oraz opisano proces konfiguracji bazy danych TimesTen jako pamięci podręcznej dla serwera bazy danych Oracle 11g.

1. Wprowadzenie

Baza danych Oracle TimesTen 11g to relacyjna baza danych zoptymalizowana pod kątem przechowywania wszystkich struktur fizycznych oraz danych tylko i wyłącznie w pamięci ulotnej RAM. Dzięki zastosowaniu specjalizowanych struktur danych i ścieżek dostępu, a także dzięki przeprojektowaniu algorytmów optymalizacji i wykonywania zapytań, TimesTen prezentuje efektywność i szybkość przetwarzania absolutnie niedostępne dla tradycyjnych, dyskowych baz danych. TimesTen jest specjalnie przystosowana do działania w warstwie aplikacji (w przeciwieństwie do tradycyjnych baz danych umiejscowionych w warstwie danych lub warstwie zasobów), w szczególności, TimesTen osiąga maksymalną efektywność w przypadku bezpośredniego działania w obszarze pamięci aplikacji z komunikacją wewnątrz-procesową, ale może także funkcjonować jako niezależny moduł, z którym aplikacje komunikują się poprzez tradycyjne interfejsy programistyczne zgodnie z architekturą klient-serwer [www02]. Zanim jednak zostaną przedstawione architektura i struktury fizyczne TimesTen, warto się zastanowić, jakie przyczyny i wymagania poskutkowały pojawieniem się tego produktu na rynku.

Tradycyjne systemy baz danych od lat były krytykowane za brak wsparcia dla przetwarzania w czasie rzeczywistym lub bliskim czasowi rzeczywistemu. Istnieją niezliczone przykłady aplikacji, dla których możliwość przetwarzania danych w czasie rzeczywistym jest absolutnie kluczowa: operatorzy telekomunikacyjni, brokerzy, firmy zajmujące się logistyką, telefoniczne centra obsługi klienta, agendy rządowe związane z bezpieczeństwem i wywiadem, ale także tradycyjne firmy adaptujące architekturę zorientowaną na usługi (ang. *SOA*, *Service Oriented Architecture*) jako podstawową architekturę systemu informatycznego. W szczególności transformacja do modelu SOA pociąga za sobą powstanie nowych, wcześniej niespotykanych wyzwań natury technologicznej, dla których idealnym rozwiązaniem są bazy danych pamięci operacyjnej.

Wiele nowoczesnych systemów informatycznych przyjmuje techniki kolejkwania wiadomości jako podstawowy mechanizm komunikacji między komponentami systemu. Związany z tym wzrost liczby wiadomości przepływających przez sieci biznesowe wymusza tworzenie narzędzi do przechwytywania, analizowania i odpowiadania na wiadomości w czasie rzeczywistym. Źródłem rosnącej liczby komunikatów w środowisku systemu informatycznego są aplikacje wykorzystujące czujniki lub znaczniki RFID, portale internetowe, usługi sieciowe uruchamiane dla zewnętrznych klientów, czy też systemy sterowania przepływem pracy, procesów i dokumentów. Wszystkie zdarzenia generowane przez wyżej wymienione systemy muszą być przetwarzane efektywnie, sprawnie, elastycznie i natychmiast. Wolumen transakcji związanych z obsługą zdarzeń jest często tak duży, że przeniesienie przetwarzania do tradycyjnej bazy danych przestaje być akceptowalnym rozwiązaniem ze względu na pojawiające się opóźnienia czasowe. Pojawia się konieczność przetwarzania zdarzeń bezpośrednio w warstwie aplikacji.

Kłopot w tym, że współczesne systemy informatyczne, oparte na tradycyjnych bazach danych, są zupełnie nieprzygotowane do spełnienia stawianych im wymogów. Pojawia się zupełnie nowa warstwa w architekturze systemu informatycznego, zwana warstwą aplikacji przedsiębiorstwa (ang. *enterprise application tier*), której zadaniem jest przetwarzanie zdarzeń biznesowych w czasie rzeczywistym. Głównym efektem wprowadzenia warstwy aplikacji przedsiębiorstwa jest gruntowna przebudowa warstwy front-end i uzbrojenie aplikacji końcowych w narzędzia do przetwarzania i błyskawicznego odpowiadania na zdarzenia biznesowe. Nie wolno jednak zapominać, że znakomita większość danych przedsiębiorstwa wciąż jest i będzie składowana w tradycyjnych bazach danych. W większości przypadków dane przedsiębiorstwa można wyraźnie rozdzielić na część statyczną, rzadko podlegającą zmianie, służącą do ogólnie rozumianych celów analityki biznesowej, oraz część dynamiczną, aktualnie wykorzystywaną i aktywną. Praktycznie zawsze wolumeny danych statycznych wielokrotnie przerastają wolumeny danych dynamicznych. W efekcie, optymalnym środowiskiem zarządzania danymi statycznymi są tradycyjne systemy baz

danych umiejscowione w warstwie zasobów, natomiast dane dynamiczne wymagają nowej, lekkiej warstwy zarządzającej w czasie rzeczywistym. Warstwa aplikacji przedsiębiorstwa ma za zadanie wypełnienie tej właśnie potrzeby.

Warto w tym miejscu nadmienić, że pierwsze próby przeniesienia przetwarzania zdarzeń biznesowych do warstwy aplikacji były już niejednokrotnie podejmowane, najczęściej pod postacią specjalizowanych aplikacji, dostrojonych starannie do modelowanej rzeczywistości. Pomijając nawet koszty i technologiczne wyzwania związane z tworzeniem takiego systemu zarządzania zdarzeniami w czasie rzeczywistym, przygotowanie takiego rozwiązania wymagało dużych nakładów czasowych i zasobowych. Poza tym, aplikacje projektowane dla określonej rzeczywistości często były mało elastyczne i zmiany w polityce funkcjonowania przedsiębiorstwa mogły pociągać za sobą daleko idące ingerencje w kod aplikacji. Ponieważ napisanie w pełni funkcjonalnego systemu zarządzania bazą danych w czasie rzeczywistym przerastały możliwości znakomitej większości potencjalnych klientów, proponowane rozwiązania w kwestii zarządzania danymi często ograniczały się do buforowania danych w warstwie aplikacji i późniejszej propagacji danych do tradycyjnej bazy danych. Nie trzeba więc wspominać, że zagadnienia związane z zarządzaniem współbieżnym dostępem, odtwarzanie stanu spójnego po awarii, tworzenie kopii bezpieczeństwa, czy optymalizacja dostępu do danych, były adresowane szczątkowo.

Rozwiązaniem większości problemów opisanych powyżej jest zastosowanie uniwersalnego środowiska zarządzania zdarzeniami biznesowymi w czasie rzeczywistym. Takim właśnie środowiskiem jest baza Oracle TimesTen 11g. Jest to relacyjny system zarządzania danymi przystosowany do przetwarzania danych i zdarzeń w pamięci operacyjnej, z całkowitym pominięciem nośników dyskowych. TimesTen jest bazą zaprojektowaną do wykorzystania bezpośrednio w warstwie aplikacyjnej, choć może również funkcjonować w architekturze klient-serwer. Zastosowanie nowoczesnych rozwiązań strukturalnych oraz daleko idące uproszczenie architektury systemu zarządzania bazą danych umożliwia osiągnięcie efektywności i przepustowości niedostępnych dla tradycyjnych baz danych. W dalszej części artykułu opisano bazę danych TimesTen, jej podstawowe struktury fizyczne, a także przedstawiono możliwości wykorzystania bazy danych TimesTen jako zaawansowanej pamięci podręcznej zsynchronizowanej z główną bazą danych w wielu różnorodnych konfiguracjach sposobów i protokołów replikacji danych.

2. Baza danych pamięci operacyjnej

Istnieje wiele produktów oferujących możliwość przetwarzania danych całkowicie w pamięci operacyjnej. Spośród najbardziej popularnych baz danych pamięci operacyjnej (ang. *IMDB, in-memory database*) wymienić należy FastDB [Fa], eXtremeDB [Ext], MonetDB [Mon], oraz Polyhedra IMDB [Pol]. IMDB to technologia zarządzania danymi, które w całości mieszczą się w pamięci ulotnej RAM. Najczęściej, bazy danych pamięci operacyjnej oferują pełne przetwarzanie transakcyjne z gwarancją zachowywania atomowości, spójności, izolacji i trwałości, umożliwiają konsolidację (ang. *linking*) bazy danych bezpośrednio z przestrzeni adresowej aplikacji bądź łączenie się za pomocą tradycyjnych interfejsów (JDBC, ODBC, lub interfejsy specyficzne dla danego dostawcy), oraz zapewniają możliwość odtworzenia stanu spójnego bazy danych po awarii. Technologia IMDB, choć może się to wydawać w pierwszej chwili sprzeczne z intuicją, prowadzi do bardzo daleko idącego uproszczenia architektury systemu baz danych. Najważniejszą cechą wszystkich systemów IMDB jest zwolnienie bardzo dużej ilości mocy obliczeniowej procesorów poprzez zaprzestanie zarządzania wieloma lokalizacjami danych.

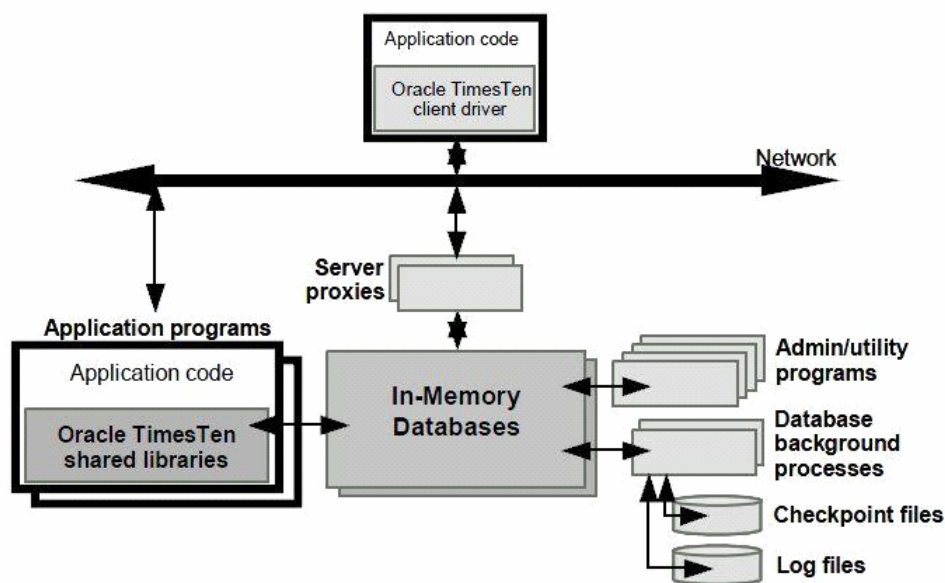
Tradycyjny system zarządzania bazą danych przechowuje wszystkie dane w blokach dyskowych, natomiast zapytania działają na wielu różnych kopiach tych samych danych umieszczonych na stronach pamięci w buforze bazy danych. Tradycyjny system zarządzania bazą danych musi poświęcić bardzo dużo pracy w celu uspoźnienia danych i synchronizacji danych między pamięcią operacyjną i dyskiem magnetycznym. Baza danych pamięci operacyjnej wykorzystuje dysk ma-

gnetyczny tylko jako medium utrwalania i składowania danych w momencie wyłączenia bazy danych, oraz do utrzymywania plików kontrolnych na potrzeby odtwarzania stanu spójnego po awarii. Brak konieczności uspoźniania stanu stron pamięci operacyjnej ze stanem bloków dyskowych ma dramatyczny wpływ na ogólną przepustowość systemu.

Pierwsze prace nad bazą danych pamięci operacyjnej zostały rozpoczęte w 1992 roku w laboratoriach badawczych HP Laboratories. W 1996 roku osoby pracujące nad projektem Smallbase opuściły HP Laboratories i utworzyły prywatną firmę TimesTen Performance Software. Pierwsza wersja bazy danych pamięci operacyjnej ujrzała światło dzienne w 1998 roku, a w 2001 roku powstał produkt TimesTen/Cache, rewolucyjne rozszerzenie pamięci podręcznej tradycyjnej bazy danych. Firma TimesTen została kupiona przez Oracle Corporation w 2005 roku, i już po dwóch latach, w 2007 roku, powstała TimesTen 7.0, pierwsza baza danych pamięci operacyjnej zbudowana od początku do końca przez Oracle i, rzecz jasna, ściśle zintegrowana z flagowymi produktami Oracle: systemem zarządzania bazą danych Oracle 10g i architekturą RAC. Aktualnie, najnowsza wersja oprogramowania to TimesTen 11g wypuszczona w lipcu 2009 roku.

2.1. Struktury fizyczne bazy danych TimesTen

Baza danych TimesTen wykorzystuje specjalizowane struktury fizyczne, zoptymalizowane pod kątem przetwarzania w pamięci operacyjnej. Ogólna architektura bazy danych TimesTen została przedstawiona na Rys. 1.

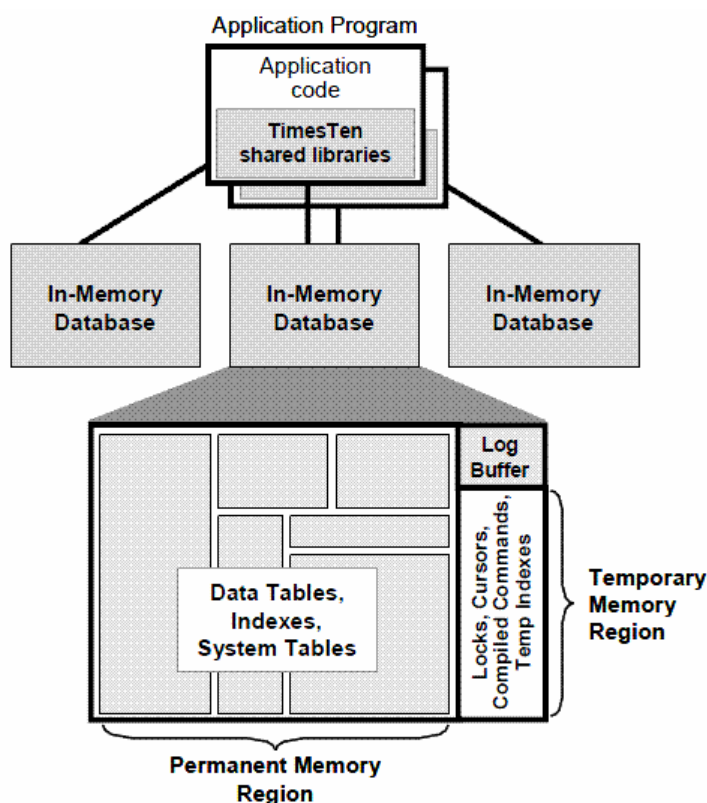


Rys. 1. Ogólna architektura bazy danych TimesTen (źródło: [www01])

Cała funkcjonalność bazy danych TimesTen jest zamknięta w zbiorze bibliotek współdzielonych (ang. *shared libraries*), które mogą być konsolidowane z aplikacją i wywoływane bezpośrednio z kodu aplikacji, współdzieląc przestrzeń adresową z aplikacją. W ten sposób baza danych TimesTen osiąga najwyższą efektywność działania, ponieważ całkowicie rezygnuje z komunikacji międzyprocesowej lub komunikacji w trybie klient-serwer. Widać tu pierwszą podstawową różnicę między bazą danych TimesTen a tradycyjną bazą danych, która najczęściej jest implementowana jako kolekcja programów uruchamianych niezależnie od aplikacji jako osobne procesy. Z uruchomieniem bazy danych bezpośrednio z kodu aplikacji wiąże się jednak bardzo poważny problem: błąd w kodzie aplikacji lub wyjątek powodujący zatrzymanie procesu aplikacji mógłby wprowadzić chaos do bazy danych i spowodować powstanie stanu niespójnego poprzez niekontrolowane wyłączenie bazy danych. Baza danych TimesTen broni się przed takim zjawiskiem po-

przez ochronę kodu umieszczonego w bibliotekach współdzielonych. Mechanizm ochronny został zaimplementowany w postaci opatentowanego algorytmu MicroLogging, który zapobiega zatrzymaniu procesu bazy danych nawet w przypadku awarii nadrzędnego procesu aplikacji. Rys. 1 pokazuje także, że oprócz bezpośredniej konsolidacji bibliotek TimesTen z kodem aplikacji można również skomunikować się z procesem TimesTen korzystając z komunikacji sieciowej. TimesTen wspiera całą gamę różnych interfejsów, umożliwiając przesyłanie poleceń SQL i PL/SQL za pomocą ODBC (jest to interfejs natywnie zaimplementowany w TimesTen), JDBC, OCI, Pro*C oraz TTCclasses, zbioru klas języka C++ stanowiących otoczkę wokół interfejsu ODBC. TimesTen umożliwia również śledzenie transakcji działających wewnątrz bazy danych poprzez interfejsy XLA (czytanie logu transakcyjnego) i JMS (usługa kolejkowania wiadomości dla platformy Java EE). Należy tu zaznaczyć, że optymalnym trybem pracy dla bazy danych TimesTen jest konsolidacja bibliotek współdzielonych bezpośrednio w warstwie aplikacji.

Struktury fizyczne do przechowywania danych zostały zilustrowane na Rys. 2. Wszystkie elementy: dane, indeksy, słownik bazy danych, bufor logu, tablice do przechowywania informacji o założonych blokadach, czy segmenty tymczasowe, są przechowywane tylko i wyłącznie w segmentach pamięci współdzielonej systemu operacyjnego. Na jednej fizycznej maszynie może być jednocześnie uruchomionych wiele baz danych TimesTen, każda baza danych może obsługiwać wiele aplikacji i każda aplikacja może jednocześnie łączyć się z wieloma bazami TimesTen. Najbardziej charakterystyczną cechą TimesTen jest brak buforów danych. Faktycznie, o ile tradycyjna baza danych musi utrzymywać w pamięci operacyjnej bufor zawierający kopie bloków dyskowych, TimesTen w ogóle nie wykorzystuje buforów danych. Wszystkie operacje są wykonywane bezpośrednio na właściwych stronach pamięci, zawierających dane, indeksy czy segmenty tymczasowe. W efekcie, kod obsługujący dane ulega znacznemu uproszczeniu, ścieżki dostępu do danych ulegają skróceniu, baza danych unika bardzo kosztownego kopiowania bloków dyskowych do pamięci, znika konieczność komunikacji z kontrolerem dysku twardego, a wszystkie algorytmy przetwarzania zapytań stają się prostsze.



Rys. 2. Struktury składające się na bazę danych TimesTen (źródło: [1])

Kolejnym elementem składowym bazy danych TimesTen są procesy systemowe. Są to procesy tła obsługujące zdarzenia uruchomienia i zatrzymania bazy danych TimesTen, wykrywanie awarii aplikacji nadrzędnej, ładowanie bazy danych z plików dyskowych do pamięci w trakcie uruchamiania bazy danych, a także procesy odpowiedzialne za wykonywanie punktów kontrolnych, zapis pliku dziennika oraz wykrywanie zakleszczenia na poziomie bazy danych. Są to kluczowe procesy z punktu widzenia zapewnienia przetwarzania transakcyjnego. Jak już wcześniej wspomniano, TimesTen wspiera pełny model przetwarzania transakcyjnego, gwarantujący atomowość, spójność, izolację i trwałość transakcji. TimesTen wykorzystuje techniki podobne do technik spotykanych w tradycyjnych bazach danych w celu umożliwienia odtworzenia bazy danych po awarii. Trwały obraz danych jest przechowywany w plikach kontrolnych (ang. *checkpoint file*) umieszczonych na dysku. Pliki kontrolne są czytane podczas uruchamiania bazy danych, a ich zawartość jest ładowana do pamięci operacyjnej. Od tego momentu specjalizowany proces tła wykonuje okresowe punkty kontrolne, w trakcie których zawartość pamięci operacyjnej jest zapisywana na dysk. Częstotliwość wykonywania punktów kontrolnych może być ustalona w oparciu o interwały czasowe lub w oparciu o sumaryczną ilość zmian, jakie zaszły w bazie danych od poprzedniego punktu kontrolnego. Punkty kontrolne są wykonywane w trybie rozmytym (ang. *fuzzy checkpoint*) w taki sposób, aby zminimalizować wpływ punktu kontrolnego na działanie całej bazy danych, choć można także wykonać punkt kontrolny spójny z transakcjami (ang. *transaction-consistent checkpoint*), który spowoduje chwilową blokadę całej bazy danych. Baza danych TimesTen może być skonfigurowana w trybie trwałym lub ulotnym. Tryb trwały umożliwia zapis obrazu pamięci do plików kontrolnych. W trybie ulotnym dane są utrzymywane tylko w czasie działania bazy danych, pliki kontrolne nie są wykorzystywane, a punkty kontrolne nie są uruchamiane.

Pomiędzy punktami kontrolnymi wszystkie zmiany wprowadzane przez transakcje, procesy replikacji i procesy synchronizacji są zapisywane w dzienniku (ang. *log*). W zależności od wybranego trybu pracy bufor dziennika mogą być zapisywane synchronicznie lub asynchronicznie. W przypadku zapisu synchronicznego umieszczenie wpisu w pliku dziennika na dysku stanowi część procesu zatwierdzania transakcji. Zapis synchroniczny gwarantuje, że żadna zmiana nie zostanie utracona i stan spójny bazy danych może być odzyskany poprzez uspójnienie poprzedniego pliku kontrolnego bazy danych z plikiem dziennika. Jeśli aplikacja wymaga maksymalnej przepustowości przetwarzania, a jednocześnie model biznesowy pozwala na utratę niektórych transakcji, baza danych TimesTen może wykonywać asynchroniczny zapis buforów dziennika. W takim przypadku wiele rekordów z bufora dziennika zostanie zapisane do pliku dziennika za pomocą jednej operacji wejścia-wyjścia. Należy się jednak liczyć z możliwością utraty informacji o niektórych transakcjach w przypadku wystąpienia awarii w trakcie zapisu bufora dziennika.

Ostatnią składową bazę danych TimesTen są aplikacje do administracji. Są to programy powłoki, wykonujące typowe czynności administracyjne: tworzenie kopii bezpieczeństwa (`ttBackup`), odtwarzanie (`ttRestore`), kopiowanie do innej instancji (`ttAdoptStores`), migracja bazy danych (`ttBulkCp`, `ttMigrate`), monitorowanie bazy danych (`ttSchema`, `ttSize`, `ttStatus`), usuwanie bazy danych (`ttDestroy`), itp. Ważnym i przydatnym narzędziem jest interaktywna konsola języka SQL (`ttiSQL`).

2.2. Przetwarzanie zapytań

Bez wątpienia, jednym z najciekawszych aspektów bazy danych TimesTen jest proces przetwarzania zapytań. Praktycznie całe niebagatelne przyspieszenie przetwarzania zapytań w bazie TimesTen wynika z prostej zmiany jednego założenia: miejsca przebywania danych w trakcie wykonywania zapytania. Dostęp do bloków dyskowych jest obciążony bardzo dużym kosztem, zaś tradycyjne bazy danych przygotowują plany wykonania zapytania nie wiedząc a priori, czy przetwarzane dane znajdują się jeszcze na dysku, skąd muszą zostać pobrane i przekopiowane do bufora, czy też może są już dostępne w buforze. W efekcie, przygotowywane plany wykonania zapytania optymalizują najgorszy możliwy scenariusz, w którym dane muszą zostać wcześniej pobrane

z dysku, a proponowane plany minimalizują liczbę odczytywanych bloków dyskowych. Taka strategia, mimo że ze wszech miar słuszna, produkuje suboptymalne plany w przypadku gdy przetwarzane dane już znajdują się w buforach danych. W przeciwieństwie do tradycyjnej bazy danych, baza danych TimesTen zakłada, że wszystkie dane potrzebne do przetworzenia zapytania znajdują się w pamięci operacyjnej i dostęp do każdego elementu danych jest swobodny, a zatem nie trzeba optymalizować najgorszego scenariusza. Co więcej, przyjmując założenie o jednakowym koszcie dostępu do dowolnego elementu danych, szacunki optymalizatora stają się prostsze i dużo bardziej dokładne.

Innym aspektem procesu przetwarzania zapytań, odróżniającym bazę danych TimesTen od tradycyjnej bazy danych, jest architektura indeksów. Indeksy pełnią w bazie danych TimesTen tę samą rolę, co w tradycyjnej bazie danych. Indeksy są wykorzystywane do przetwarzania wszystkich poleceń SELECT, INSERT, UPDATE i DELETE, do wyszukiwania zakresowego, sortowania, grupowania i łączenia tabel. Mimo, że dostęp do elementów danych jest swobodny, indeksy pozwalają uniknąć żmudnego przeglądania liniowego całej pamięci. TimesTen oferuje trzy typy indeksów: indeks zakresowy, indeks haszowy i indeks bitmapowy. Jedyną nowość stanowi indeks zakresowy, przyspieszający zapytania o wartości zawierające się w podanym zakresie. Podczas łączenia tabel optymalizator bazy danych TimesTen rozważa dwa algorytmy łączenia: połączenie typu *nested-loop* albo połączenie typu *sort-merge* (w ostatnim przypadku bardzo przydatne okazują się indeksy zakresowe).

Transakcje wykonywane w bazie danych TimesTen mogą być uruchamiane współbieżnie. Dopuszczalne poziomy izolacji to *serializable* i *read-committed*. Transakcje działające w trybie *read-committed* nie widzą zmian wprowadzanych przez inne, niezatwierdzone transakcje. Jeśli transakcja działa w trybie *serializable*, wówczas transakcja nie widzi żadnych zmian wprowadzonych do bazy danych przez jakąkolwiek transakcję. Do zapewnienia powyższych poziomów izolacji współbieżnych transakcji, baza danych TimesTen wykorzystuje mechanizm wielowersyjności oraz mechanizm blokad. Blokady mogą być zakładane na poziomie pojedynczego wiersza, całej tabeli bądź całej bazy danych. Blokady dotyczą tylko i wyłącznie operacji zapisu, operacje odczytu nie są nigdy blokowane (jedyny wyjątek dotyczy operacji odświeżającej perspektywę materializowaną). Jak już wcześniej wspomniano, transakcje mogą się wykonywać z pełnym wsparciem cech ACID, ale można także połuźnić kryterium trwałości poprzez asynchroniczne, leniwe zapisywanie buforów dziennika do pliku dziennika.

Baza danych TimesTen może być replikowana w celu zwiększenia niezawodności i przepustowości systemu. Replikacja polega na równoległym utrzymywaniu wielu kopii tej samej bazy danych wraz z propagacją zmian między bazami danych. Replikacja baz danych TimesTen odbywa się na podstawie modelu *master/subscriber*, w którym jeden węzeł jest desygnowany jako *master* (modyfikacje dotyczą tylko tego węzła), a drugi jest desygnowany jako *subscriber*. Replikowane węzły komunikują się za pomocą protokołu TCP/IP przez agentów replikacji, którzy są zobowiązani do okresowego przeglądania dziennika i przesyłania informacji o zmianach, które zaszły w każdej z replikowanych baz danych. Domyślnie, replikacja odbywa się asynchronicznie, lecz środowisko dostarcza także dwóch dodatkowych usług, które można włączyć wraz z replikacją synchroniczną:

- *powiadomienie*: aplikacja zostaje zablokowana do momentu, w którym węzeł subskrybujący nie potwierdzi otrzymania informacji o zmianie,
- *potwierdzenie*: aplikacja zostaje zablokowana do momentu, w którym węzeł subskrybujący nie potwierdzi otrzymania i zatwierdzenia informacji o zmianie.

Bazy danych TimesTen mogą być replikowane w dwóch konfiguracjach: *active-active* i *active-standby*, przy czym druga konfiguracja jest bardzo mocno zalecana przez producenta. W przypadku konfiguracji *active-standby*, tylko aktywna baza danych może podlegać modyfikacjom. Wszystkie zmiany wprowadzone do bazy *active*, są następnie replikowane (synchronicznie lub

asynchronicznie) do bazy *standby*. Stąd mogą być już dalej propagowane (np. przez propagację fragmentu pliku dziennika) do dalszych baz danych. W dowolnym momencie może funkcjonować tylko jedna baza danych *master*. Jeśli ulegnie ona awarii, natychmiast jedna z baz *standby* zostaje awansowana do bazy *master*.

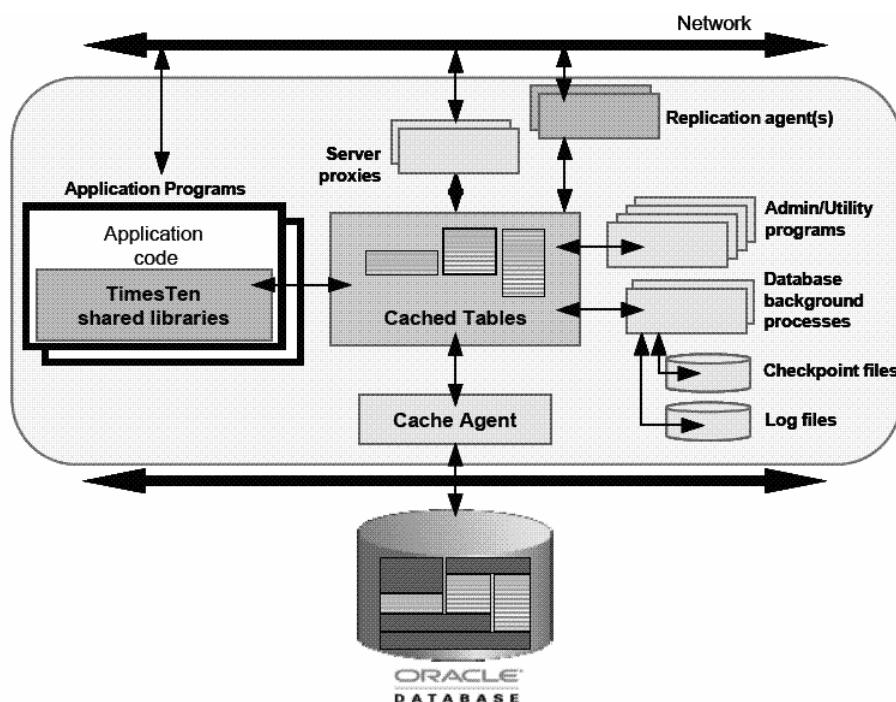
3. Pamięć podręczna bazy danych

Jak już wcześniej wspomniano, problem buforowania danych w warstwie aplikacji był rozważany od dawna i opracowano wiele schematów buforowania, które miały zaspokajać różnorodne potrzeby aplikacji. Wszystkie poniższe schematy mają na celu przeniesienie danych jak najbliżej aplikacji w celu minimalizacji narzutu czasowego związanego z komunikacją, uproszczenie aplikacji, oraz zmniejszenie obciążenia bazy danych przez lokalne przetwarzanie przynajmniej wybranych zapytań. Można wyróżnić następujące schematy buforowania:

- *buforowanie wyników zapytań*: wykonywane w warstwie aplikacji przez specjalizowane oprogramowanie, sprowadza się do zapamiętania wyników ostatnio zgłoszonych zapytań i przechwycenie tych zapytań, które mogą być obsłużone na podstawie pamięci podręcznej, jest to technika prosta, która dobrze sprawdza się w przypadku często powtarzających się zapytań,
- *buforowanie w warstwie odwzorowania obiektowo-relacyjnego*: wykorzystuje wewnętrzne mechanizmy technik odwzorowania obiektowo-relacyjnego (np. Hibernate lub JPA) do buforowania zmian w warstwie aplikacji i okresowej synchronizacji bazy danych,
- *buforowanie obiektów*: repozytorium obiektów, nad którymi aplikacja ma pełną programistyczną kontrolę, funkcjonują jak obiektowe bazy danych.

TimesTen oferuje możliwość skonstruowania w pełni funkcjonującego, rozproszonego środowiska działającego jak pamięć podręczna do większej dyskowej bazy danych. Produkt ten nazywa się Oracle In-Memory Database Cache (*IMDB Cache*) i może występować zarówno w postaci pojedynczej bazy danych TimesTen stanowiącej pamięć podręczną dla tradycyjnej relacyjnej bazy danych, jak i w postaci siatki pamięci podręcznych (ang. *Cache grid*) składającej się z powiązanych ze sobą baz danych, oferujących aplikacjom transparentność lokalizacji danych. Wykorzystanie siatki pamięci podręcznych zdecydowanie poprawia nie tylko wydajność systemu bazy danych, ale umożliwia elastyczny rozwój i sukcesywne podnoszenie wydajności poprzez dodawanie nowych węzłów do siatki. Obecność bazy danych TimesTen jako pamięci podręcznej dla tradycyjnej bazy danych poprawia wydajność dwojako: zdecydowanie zmniejsza obciążenie systemu baz danych poprzez przejęcie przetwarzania większości zapytań, oraz skraca czas oczekiwania aplikacji na dane poprzez eliminację czasochłonnej komunikacji sieciowej.

IMDB Cache przechowuje lokalnie kopie tabel ze źródłowej bazy danych. Kopie tych tabel są lokalnie modyfikowalne (a zatem pamięć podręczna jest pamięcią modyfikowalną, a nie tylko pamięcią do odczytu), a zmiany są automatycznie synchronizowane z oryginalną bazą danych. Rys. 3. ilustruje ogólną architekturę IMDB Cache.



Rys. 3. Architektura IMDB Cache (źródło: [www02])

Zawartość pamięci podręcznej jest definiowana na podstawie danych przechowywanych w tradycyjnej dyskowej bazie danych. Tabele, przechowywane w bazach TimesTen powiązanych z daną bazą danych, tworzą grupy pamięci podręcznej (ang. *cache groups*), przy czym tabele te muszą być obowiązkowo powiązane w oryginalnej bazie danych za pomocą ograniczeń referencyjnych. Zbiór krotek powiązanych ograniczeniami referencyjnymi jest nazywany instancją pamięci podręcznej (ang. *cache instance*). Każda instancja może być ładowana ze źródłowej bazy danych do pamięci podręcznej bądź jawnie, bądź dynamicznie. Jawne ładowanie może dotyczyć całej grupy pamięci podręcznej, grupy instancji spełniających kryteria wyszukiwania wyspecyfikowane w klauzuli WHERE, oraz grupy instancji wybranych wg. Identyfikatora. Ładowanie dynamiczne polega na tym, że w przypadku „pudła” (poszukiwana krotka nie została znaleziona w pamięci TimesTen), odpowiednie krotki powiązane także zostały załadowane do pracy. Z problemem ładowania pamięci podręcznej nieodłącznie wiąże się problem zwalniania pamięci podręcznej. TimesTen oferuje dwa kryteria wyboru instancji, które zostaną usunięte z pamięci podręcznej: postarzanie wg. ostatniego czasu użycia (ang. *usage-based aging*) oraz postarzanie wg. czasu (ang. *time-based aging*).

TimesTen umożliwia tworzenie dwóch rodzajów grup pamięci podręcznej. Grupy lokalne zachowują autonomię co do składowanych danych i nie replikują swoich danych pomiędzy innymi węzłami wchodzącymi w skład grupy. W efekcie nie występuje transparentność lokalizacji danych i aplikacja ma za zadanie zarządzanie dostępem do właściwej bazy danych. Globalne grupy pamięci podręcznej tworzą rozproszoną pamięć podręczną i współdzielą fragmenty buforowanych tabel. Zarówno grupy lokalne, jak i globalne, mogą być modyfikowalne, choć preferowany jest wariant z grupami pamięci podręcznej tylko do odczytu. Grupy pamięci podręcznej mogą być zarządzane przez system lub przez użytkownika. Dla grup zarządzanych przez system można wyróżnić następujące trzy rodzaje:

- grupy pamięci podręcznej tylko do odczytu (ang. *read-only cache groups*): modyfikacje możliwe tylko w bazie źródłowej i propagowane do pamięci podręcznej,

- grupy pamięci podręcznej z zapisem asynchronicznym (ang. *asynchronous writethrough cache group*): modyfikacje możliwe tylko w pamięci podręcznej, propagowane do bazy źródłowej asynchronicznie po zakończeniu transakcji,
- grupy pamięci podręcznej z zapisem synchronicznym (ang. *synchronous writethrough cache group*): modyfikacje możliwe tylko w pamięci podręcznej, propagowane do bazy źródłowej synchronicznie po zakończeniu transakcji,

Jeśli chodzi o sposoby synchronizacji pamięci podręcznej ze źródłową bazą danych, to TimesTen oferuje dwa tryby synchronizacji:

- *propagate*: zmiany w pamięci podręcznej są indywidualnie przesyłane do bazy danych,
- *flush*: krotki przechowywane w bazie danych zostają nadpisane aktualną zawartością pamięci podręcznej.

Synchronizacja odwrotna, z bazy danych do pamięci podręcznej, może być wykonywana na trzy sposoby:

- *refresh*: odpowiednik wyładowania i załadowania pamięci podręcznej,
- *full autorefresh*: umożliwia wskazanie interwałów czasowych określających częstotliwość odświeżenia całej zawartości bazy danych,
- *incremental autorefresh*: odświeża tylko te instancje pamięci podręcznej, które uległy zmianie w trakcie ostatnich 3 miesięcy.

4. Podsumowanie

Wzrost popularności architektury SOA powoduje gwałtowny przyrost komunikatów i zdarzeń biznesowych, które muszą być obsługane przez system informatyczny w czasie rzeczywistym. Istniejąca infrastruktura informatyczna nie jest przygotowana do takich wyzwań, stąd pojawia się konieczność znalezienia lekkich, elastycznych narzędzi które mogą przetwarzać komunikaty i zdarzenia w warstwie aplikacji w sposób efektywny, sumienny i nowoczesny.

Oracle TimesTen to baza danych pamięci operacyjnej, która została specjalnie zaprojektowana jako wbudowana baza danych pamięci operacyjnej. TimesTen oferuje błyskawiczne przetwarzanie danych i zdarzeń w warstwie aplikacji przedsiębiorstwa, automatyzując wiele składowych procesów przetwarzania zdarzeń. Jest to nowoczesny, interesujący produkt, który bez wątpienia szybko znajdzie szeroką rzeszę entuzjastów.

Bibliografia

- [Ext] <http://www.mcobject.com/extremedbfamily.shtml>
- [Fa] <http://www.garret.ru/fastdb.html> Main Memory Relational Database Management System
- [Mon] <http://www.monetdb.nl/>, Query Processing at Light Speed
- [OTWP09a] Extreme Performance Using Oracle TimesTen, Oracle Technical White Paper, July 2009
- [OTWP09b] Using Oracle In-Memory Database Cache to Accelerate the Oracle Database, Oracle Technical White Paper, July 2009
- [Pol] <http://www.downloaddatabase.com/databasemanagement/polyhedra.html>
- [www01] http://it.toolbox.com/wiki/index.php/Oracle_TimesTen
- [www02] http://download.oracle.com/docs/cd/E13085_01/welcome.html Oracle TimesTen In-Memory Database Documentation