Platforma Caché (II)

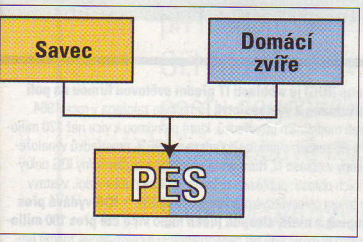
MAREK KOCAN

V dnešním pokračování se blíže podíváme na další vlastnosti platformy Caché od společnosti InterSystems.

Bitmapy v databázích

Bitmapové indexy jsou dobře známé a využitelné z OLAP systémů lze si je představit například jako sady předem vypočítaných odpovědí na jednoduché otázky. Bitmapové indexy, na rozdíl od klasických indexů, pracují se dvěma hodnotami – pravdou a nepravdou. Zjednodušeně tedy bitmapový index říká, zda je odpověď na danou otázku pro ten který záznam v databázi pravdivá či nikoliv.

Využitelnost je prakticky všude tam, kde běžné indexové metody selhávají – například pro několik málo číselníkových hodnot e vlastních datech (0 – Student, 1 – Zaměstnanec, 2 – Důchodce apod.). Další výhodou bitmapových indexů je možnost jejich efektivní komprese. V praxi totiž dochází k velice častému opakování dlouhých úseků složených buď z binárních nul nebo jedniček.



Podtřída může dědit vlastnosti od jedné nebo více nadřazených tříd.

Databázová platforma Caché umožňuje nejen klasické použití bitmapových indexů, jakožto součásti relační databáze, ale také programové vytváření bitmapových indexů. Pomocí prostředků skriptovacích jazyků ObjectScript nebo Caché Basic lze naprogramovat, jak má bitmapové pole pro nějakou podmínku vytvářet v případě, že klasický bitmapový index nelze efektivně použít. Oba skriptovací jazyky poskytují nástroje pro práci s bitovými datovými typy a pro bitové logické operace. Způsob implementace bitmapových indexů v Caché navíc umožňuje binárně indexovat i dynamicky se měnící transakční data, a to beze ztráty rychlosti transakčního zpracování – tato technologie je tak použitelná i v OLTP systémech.

Bitmapové indexy jsou v Caché ukládány jako řetězce, kde každý bit odpovídá jedné instanci a jednotlivá bitová pole se skládají z úseků dlouhých 64 tisíc bitů náležejících jednomu uzlu v globálu, v němž jsou uloženy indexy dané třídy/tabulky.

Rychlé transakce

Jednou z důležitých vlastností každé databázové platformy je snaha o odstranění úzkých hrdel a podpora transakčního zpracování. Značným problémem je obvykle velký počet uživatelů, již od stovek výše. K největším konfliktům pochopitelně dochází mezi transakcemi, které požadují přístup ke stejným datům – tedy u konkurenčních uživatelů.

Procesy databázové platformy Caché nezamykají během aktualizace dat celé stránky, ale data jsou zamykána na logické úrovni. Možné konflikty v databázi jsou dále snižovány pomocí atomizovaných operací sčítání a odčítání, které nevyžadují zamykání – například u sekvencí či generátorů identifikačních čísel.

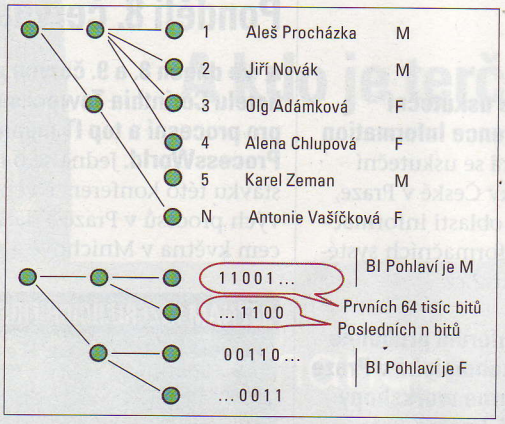
Uložení dle libosti

Zatímco k zobrazení dat v klasických dvourozměrných polích je nejvhodnější tabulkový pohled, k vypodobnění vícerozměrných datových struktur je vhodný pohled stromový. Při překladu návrhu třídy jádro Caché vytváří kód, který zajistí přístup k fyzické datové vrstvě. Programátor se tedy vůbec o tuto vrstvu nemusí zajímat a využívá data tak, jak opravdu v dané aplikaci potřebuje.

Odlišností od mnoha ostatních databázových platforem ovšem je, že v Caché mohou vývojáři sami definovat způsob uložení dat ve fyzické vrstvě. Díky této otevřenosti lze provádět optimalizaci a modifikaci uložení v těch případech, kdy původní zvolený způsob není pro náročnost aplikace optimální.

Jde to i jinam…

Brána Caché Relational Gateway umožňuje odeslat dotaz SQL vzniklý v Caché ke zpracování v jiné (relační) databázové platformě. Prostřednictvím této brány může Caché načíst a aktualizovat data uložená ve většině dostupných platforem. Pokud jsou tedy databázové třídy Caché překládány s možností CachéSQLStorage, mohou aplikace Caché přes tuto bránu používat i konkurenční prostředí a bez větších problémů tak představovat i spojovací mezičlánek ve složitějších podnikových informačních architekturách.



Využití bitmapových indexů (BI). V první části je znázorněno uložení dat, ve druhé bitmapové indexy, které jsou ukládány jako řetězce, kde každý bit odpovídá jedné instanci. Hodnoty jednotlivých bitů odpovídají konkrétní hodnotě indexu, v tomto případě muži či ženě. V prvním případě jsou pod 1 uvedeni muži (jde o BI pro pohlaví Muž), ve druhé části pod 1 ženy (jde o BI pro pohlaví Žena). Jednotlivé úseky BI přísluší odpovídajícímu uzlu v globálu, viz předchozí díl seriálu.

Mějme třídy

Při návrhu definice třídy má vývojář možnost použít průvodce pro jednotlivé dílčí úkoly nebo může psát definici přímo pomocí jazyka Class Definition Language. Těla metod mohou být napsána s využitím skriptovacích jazyků pro práci s daty a pro provádění zajištění aplikační logiky.

Objektová technologie implementovaná databázovou platformou Caché zahrnuje abstraktní třídy, registrované třídy, perzistentní, vnořené či odvozené třídy a datové typy. Abstraktní třídy slouží k popisu obecných tříd, od nichž se v rámci aplikace odvozují třídy představující konkrétní modelované objekty. K abstraktním třídám nelze vytvářet instance objektů. Registrované třídy implementují plnohodnotné objekty ve smyslu objektových technologií. Vzhledem k tomu, že nepodporují perzistenci, využívají se zpravidla pro tvorbu kódu aplikační logiky.

Perzistentní třídy slouží k popisu dat, která jsou trvale k dispozici, tedy dat ukládaných na disk nebo jiné záznamové médium. Každá perzistentní třída je odvozena od třídy registrované, při překladu je vytvářen kód, který zajišťuje komunikaci mezi aplikační vrstvou (objekt) a vrstvou fyzickou (globály).

Vnořené třídy jsou podobné třídám perzistentním, ale na rozdíl od nich nemají vlastní úložiště dat. Data v nich obsažená jsou vždy uložena spolu s daty perzistentní třídy, která na vnořenou třídu odkazuje. Odvozené třídy jsou obecně ty, jež jsou odvozené od jedné nebo více uživatelských tříd.

Datové typy jsou třídy sloužící pro popis dat a mohou být libovolně složité. Úlohou datových typů je mimo jiné zajistit správnou konverzi dat mezi zobrazovanou (externí) podobou a jejich uloženou (vnitřní) podobou či zaručit validaci vstupních hodnot.

A co vlastnosti?

Vlastnosti tříd mohou v Caché být datový typ, jednoduchá reference na jinou třídu, jedna strana relace typu: 1-N. Vlastnosti mohou být privátní, vyžadující neprázdnou hodnotu, indexované, vyžadující unikátní hodnotu, tranzientní, vypočítané a vícerozměrné. Vlastnost může být označena jako finální, takovou pak nelze v odvozených třídách modifikovat. Metody mohou být metodami instance nebo tříd. Je-li metoda metodou třídy, může být zároveň označena jako uložená procedura SQL. Metody také mohou být, podobně jako vlastnosti, privátní a finální.

(mar) 9 0186

Seriál vzniká ve spolupráci s Databázovým světem, informačním portálem o databázových technologiích (www.dbsvet.cz).