

Karol TOMASZEWSKI

Instytut Systemów Informatycznych, Wydział Cybernetyki WAT

ul. S. Kaliskiego 2, 00-908 Warszawa

E-mail: ktomaszewski@wat.edu.pl

Symulacyjna metoda badania procesu obsługi zadań dostępu do danych w OODB

1 Wprowadzenie

Obiektowe bazy danych wnoszą szereg nowych funkcjonalności do współczesnych systemów baz danych. Rozszerzenie możliwości baz obiektowych wynika między innymi ze złożoności modelu danych. Wpływa to jednak na zwiększenie złożoności całego systemu zarządzania bazą danych. Wiąże się to również ze zwiększonym zapotrzebowaniem na zasoby obliczeniowe potrzebne do przetwarzania operacji bazodanowych. Złożoność procesów przetwarzania obniża jednocześnie wydajność bazy danych w sensie czasu obsługi ządań napływających od użytkowników. Dlatego, uzasadnione jest badanie i poprawa technik wpływających na wydajność działania tych systemów.

W literaturze można znaleźć opis wielu kierunków rozwoju technik mających na celu poprawę efektywności obiektowych bazy danych.

W pracy [1] scharakteryzowane zostały wybrane zagadnienia, będące obecnie przedmiotem badań, między innymi klasteryzacja, replikacja czy współbieżność w obiektowych bazach danych.

W dalszej części pracy rozważany jest proces materializacji obiektów, tzn. proces obsługi ządań dostępu do obiektów, realizowany przez system zarządzania bazą danych.

Funkcjonalność obsługi ządań dostępu do obiektów stanowi fragment wszystkich funkcjonalności systemu zarządzania bazą danych. Na potrzeby tej pracy i dalszej analizy fragment ten zostanie uproszczony tak, aby logicznie odzwierciedlał realizowaną funkcjonalność, bez dokładnego odzwierciedlenia fizycznej implementacji. Umożliwi to sformułowanie abstrakcyjnego modelu symulacyjnego, wykorzystanego w dalszej analizie.

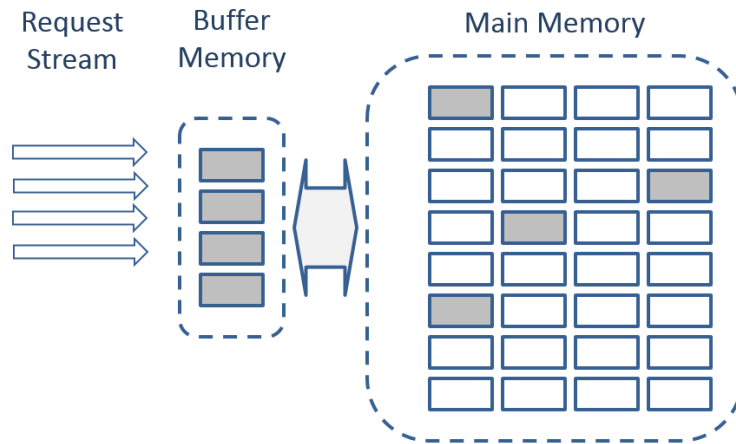
2 Proces obsługi ządań

Jedną z cech charakterystycznych, wprowadzoną do obiektowych baz danych, jest nadanie obiektom tożsamości [2]. Jest to odejście od koncepcji powiązań relacyjnych, stosowanych dotychczas w systemach relacyjnych baz danych. W relacyjnym modelu danych powiązania między danymi realizowane są z wykorzystaniem relacji kluczy głównych (*primary key*) oraz kluczy obcych (*foreign key*). Z kolei w obiektowym modelu danych powiązania między danymi realizowane są poprzez atrybuty referencyjne w definicjach obiektów. Referencje z kolei oparte są na tzw. tożsamości obiektów. Tożsamość realizowana jest przez nadanie każdej instancji obiektu identyfikatora *OID*, unikalnego w obrębie całej bazy danych. W trakcie nawigacji

obiektovej wykorzystywane są atrybuty referencyjne obiektów, odwołujące się do kolejnych instancji na podstawie referencji (identyfikatora *OID*) [2].

Praca z danymi zorganizowanymi obiektowo powoduje napływanie do obiektowej bazy danych szeregu odwołań do kolejnych obiektów, według ich tożsamości.

Typowa architektura systemów bazodanowych zakłada składowanie danych w pamięci zasadniczej (np. w pamięci dyskowej) oraz istnienie bufora, jako mniejszego obszaru w pamięci operacyjnej, odpowiadającego za obsługę bieżących żądań [3].



Rys. 1. Model „pamięć podręczna-pamięć zasadnicza”

Fig. 1. Model „buffer memory -main memory”

Każde napływające żądanie dostępu do obiektu powoduje sprawdzenie, czy żądany obiekt jest przechowywany w pamięci podręcznej - buforze. Jeżeli tak, obiekt jest udostępniany na potrzeby realizacji bieżącej transakcji. Jeżeli obiekt nie został wcześniej załadowany do pamięci podręcznej, następuje załadowanie obiektu z pamięci zasadniczej do bufora (rys. 1).

Inną istotną cechą składowania danych w systemach bazodanowych jest zgrupowanie danych w większe jednostki logiczne – stronice [4]. Podczas obsługi żądania dostępu do wybranego obiektu ładowana jest do pamięci podręcznej cała jednostka logiczna (stronica), w której obiekt był zawarty. Wiąże się to również z koniecznością algorytmicznego ustalenia innej stronicy, podlegającej usunięciu z pamięci podręcznej, oraz zastąpienie jej nową [5]. Od tego momentu żądany obiekt oraz wszystkie obiekty w obrębie tej samej stronicy są dostępne w pamięci podręcznej.

3 Rola klasteryzacji

Klasteryzacja to szeroki zakres wiedzy, wyprzedzający powstanie pierwszych komputerów. Współczesne definicje klasteryzacji nawiązują najczęściej do zagadnień analizy skupień oraz skutecznego wyznaczania zgrupowań bytów tematycznie ze sobą powiązanych. Analiza skupień opiera się z kolei na algorytmach klasyfikacji

statystycznej (bezwzorcowej), które przyporządkowują obserwacje statystyczne do zidentyfikowanych klas, bazując na atrybutach (cechach) tych obserwacji.

Wspomniana specyfika składowania obiektów, zgrupowanych w większe jednostki logiczne, umożliwia rozważenie wykorzystania informacji z analizy skupisk w celu poprawy działania procesu obsługi żądań dostępu do obiektów.

Każde odwołanie do obiektu, który nie znajduje się aktualnie w buforze, wiąże się z dodatkowym kosztem obliczeniowym i czasowym. Koszt ten wynika z szeregu operacji wykonywanych podczas ładowania obiektu z pamięci zasadniczej. Jest to między innymi odszukanie stronicy z żądanym obiektem, proces usunięcia stronicy aktualnie znajdującej się w buforze (w tym aktualizacja zmian) oraz załadowanie nowej stronicy do bufora [6].

Rozważanym problemem jest także rozmieszczenie obiektów w stronicach, aby zminimalizować ilość odwołań do kolejnych stronic. Inaczej mówiąc, należy dążyć do zminimalizowania liczby operacji wymiany stronic wykorzystywanych do obsługi napływających zapytań. W tym celu można rozważyć wykorzystanie wybranych metod klasteryzacji w organizacji bazy danych.

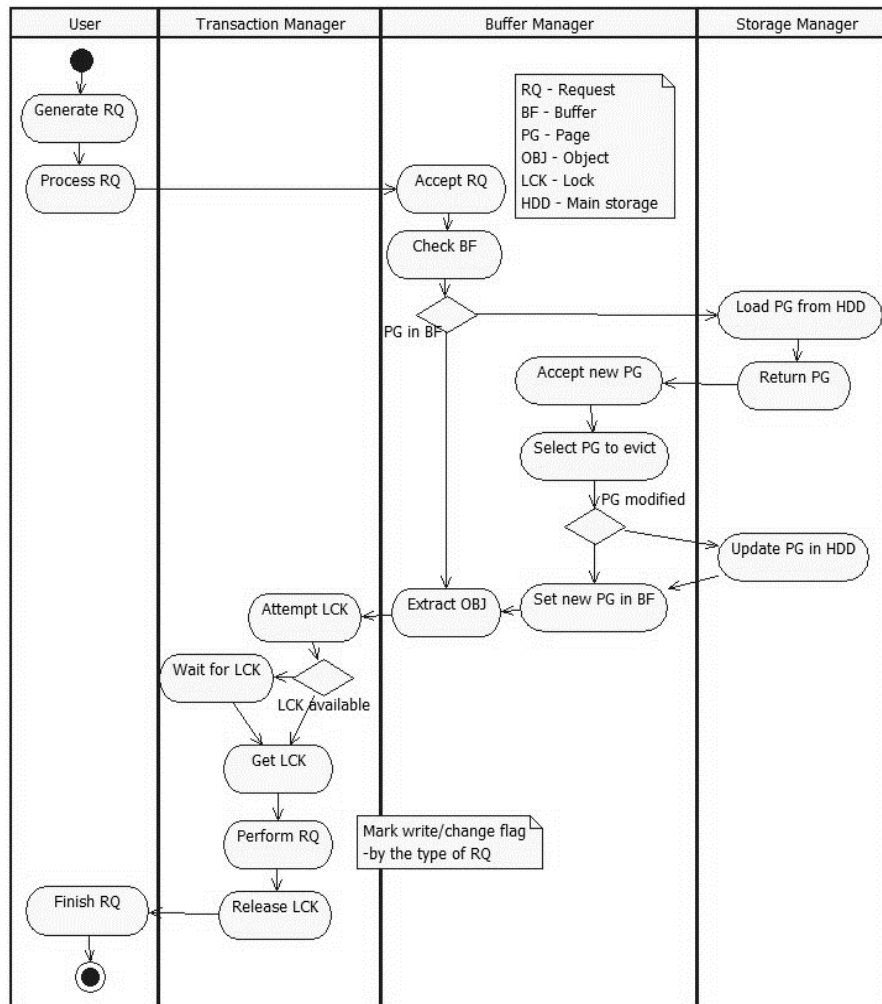
4 Symulacja

Do obserwacji pracy mechanizmu bufora można wykorzystać symulację komputerową. W symulacji takiej można rozważyć dane wejściowe jako strumień żądań dostępu do obiektów. Rolą symulowanego systemu zarządzania bazą danych jest zapewnienie dostępu do obiektu w ramach obsługi żądania. Symulowany system powinien być zorientowany według wspomnianego wcześniej modelu „pamięć podręczna-pamięć zasadnicza”. W ramach obsługi żądania obiekt jest udostępniany, jeżeli znajduje się w danej chwili w pamięci podręcznej. W przeciwnym wypadku udostępnienie obiektu wymaga odwołania do pamięci zasadniczej oraz wymiany stron w buforze [6].

Symulacja powinna uwzględniać takie charakterystyki systemów rzeczywistych, jak czas przeszukiwania zbiorów danych w pamięci podręcznej oraz czas dostępu i przeszukiwania zbiorów danych w pamięci zasadniczej.

W trakcie symulacji powinny zostać zaobserwowane takie charakterystyki, jak całkowita liczba obsłużonych żądań dostępu, liczba żądań obsłużonych tylko z wykorzystaniem pamięci podręcznej, liczba odwołań do pamięci zasadniczej oraz symulacyjny czas obsługi żądań.

Eksperyment symulacyjny można przeprowadzić, badając wykorzystanie różnych metod klasteryzacji. Różna organizacja danych w bazie powinna powodować różną trafność odwołań do pamięci. Wymienione wyżej charakterystyki powinny pozwalać ocenić i porównać wpływ wybranych metod klasteryzacji na efektywność obsługi żądań w sensie złożoności przetwarzania (liczba zdarzeń), jak i czasu przetwarzania żądań dostępu.



Rys. 2. Diagram aktywności modelu symulacyjnego

Fig. 2. Simulation model activity diagram

5 Model Symulacyjny

Model symulacyjny powinien odzwierciedlać cechy procesu udostępniania danych w systemach baz obiektowych. W szczególności powinny zostać odwzorowane zdarzenia wpływające na wydajność obsługi napływających żądań, takie jak dostęp do danych w buforze oraz wczytywanie danych z pamięci zasadniczej. Można zidentyfikować następujące encje (uczestników) wraz z zadaniami, za które

odpowiadają (rys. 2). Encja *User* jest to encja odpowiadająca za generację żądań dostępu do obiektów. Symuluje ona zdarzenie żądania dostępu do systemu bazy danych, z rozróżnieniem operacji tylko do odczytu, oraz operacji modyfikacji danych.

Przyjęte żądanie jest przekazywane do encji *Buffer Managera*. Jest to podstawowa encja, odpowiadająca za obsługę zgłoszenia. Po przyjęciu żądania następuje tu sprawdzenie, czy stronica zawierająca obiekt jest aktualnie załadowana do pamięci podręcznej. Jeżeli tak, następuje tu ekstrakcja obiektu ze stronnicy, na której obiekt ten się znajduje. Jeżeli pamięć podręczna nie zawiera stronnicy z żądanym obiektem, następuje odwołanie do kolejnej encji, tj. *Storage Managera*. Encja ta akceptuje żądania załadowania stronnicy z pamięci zasadniczej. Żądana stronica jest zwracana w ramach odpowiedzi na żądanie. Po zwróceniu stronnicy *Buffer Manager* przyjmuje nową stronicę, ustalając jednocześnie, która z bieżących stronic w pamięci podręcznej podlegać będzie usunięciu. W tym momencie następuje sprawdzenie, czy obiekty na stronie, podlegającej usunięciu z pamięci podręcznej uległy modyfikacji. Jeżeli tak, zostaje wygenerowane żądanie do encji *Storage Managera*, o zaktualizowanie stanu stronnicy w pamięci zasadniczej. Bieżąca stronica zostaje zastąpiona w pamięci podręcznej nową stronicą. Gdy potrzebne dane znajdują się już w buforze, następuje ekstrakcja obiektu z nowowczytanej stronnicy. W kolejnym kroku encja *Transaction Managera* przejmuje zarządzanie blokowaniem i dostępem do obiektu. Encja ta, po uzyskaniu dostępu do obiektu, podejmuje próbę zablokowania tego dostępu na potrzeby bieżącej transakcji. Jeżeli obiekt jest wykorzystywany w ramach innej transakcji, bieżąca aktywność zostaje zatrzymana w oczekiwaniu na zwolnienie blokady. Gdy to możliwe, bieżąca transakcja dokonuje blokady obiektu na potrzeby własnej realizacji, a następnie symulowane jest zdarzenie właściwej operacji na obiekcie. W trakcie tego zdarzenia może zostać zaznaczona flaga modyfikacji dla stronnicy, na której znajduje się obiekt. Zależy to od typu operacji w obsługiwanym żądaniu. Po tej operacji następuje zwolnienie blokady na obiekcie, po czym bieżąca transakcja jest zakończona. Zdarzenie to kończy proces obsługi bieżącego żądania.

6 Podsumowanie

W pracy tej poruszony został problem rosnącej złożoności przetwarzania, wynikającej z nowych funkcjonalności obiektowych baz danych. Rosnąca złożoność obniża efektywność przetwarzania. W artykule zaproponowana została symulacyjna metoda weryfikacji metod klasteryzacji danych na etapie poprzedzającym implementację w rzeczywistych systemach bazodanowych. Na potrzeby eksperymentu symulacyjnego zaproponowany został również model symulacyjny odzwierciedlający rzeczywisty fragment takiego systemu bazodanowego. Omawiany model ograniczony został do wybranego obszaru zainteresowań, tzn. do procesu materializacji referencji, w ramach obsługi żądań dostępu do obiektów. Model został zorientowany zdarzeniowo, odwzorowując kolejne zdarzenia procesu obsługi. Tak sformułowany model symulacyjny może pozwolić na obserwację wpływu metod klasteryzacji na działanie bufora w modelu „pamięć podręczna-pamięć zasadnicza”. Zastosowanie metod symulacyjnych pozwala na przeprowadzenie badań porównawczych i ocenę wybranych metod klasteryzacji. Można również rozważyć wykorzystanie takiego modelu symulacyjnego do badania innych technik efektywnościowych w bazach danych. Badanie zasadności metod na etapie poprzedzającym właściwą implementację

ogranicza czas i koszty prac projektowych. Pozwala również na identyfikację i wykluczanie ewentualnych błędów koncepcyjnych, niewychwyconych na etapie projektowania. Na tym etapie możliwe jest dokonywanie wyboru właściwych rozwiązań do dalszej implementacji.

Literatura

1. Tomaszewski K.: Obiektowe bazy danych – wybrane kierunki rozwoju. *Biuletyn Instytutu Systemów Informatycznych*, nr 10, 53-61, 2012
2. Barry D., Cattell R.: *The Object Data Standard: ODMG 3.0*. Morgan Kaufmann Publishers, 2000
3. Johnson T., Shasha D.: 2Q: A Low Overhead High Performance Buffer Management Replacement Algorithm. *Proceedings of the International Conference on Very Large Databases*, 439–450, 1994
4. Wang Q., Maier D., Shapiro L.: The Hybrid Technique for Reference Materialization in Object Query Processing. *Proceedings of the 2000 International Symposium on Database Engineering & Applications*, 37-46, 2000
5. He Z., Marquez A.: Path and cache conscious prefetching (PCCP). *The VLDB Journal — The International Journal on Very Large Data Bases*, nr 16, 235-249, 2007
6. Darmont J., Schneider M.: VOODB: A Generic Discrete-Event Random Simulation Model To Evaluate the Performances of OODBs. *Proceedings of the 25th International Conference on Very Large Data Bases*, 254-265, 1999

Streszczenie

W artykule omówiona została symulacyjna metoda badania wydajności procesu obsługi ządań dostępu w obiektowych bazach danych. Zaproponowany został model symulacyjny umożliwiający badanie wpływu metod klasteryzacji na efektywność omawianego procesu. Rozpatrywana jest organizacja danych w modelu „pamięć podręczna-pamięć zasadnicza”. Eksperyment symulacyjny może posłużyć do porównania wybranych metod klasteryzacji na etapie projektowania rozwiązań. Wyniki badań mogą być wykorzystane przy podejmowaniu decyzji o implementacji metod w rzeczywistych systemach bazodanowych.

Słowa kluczowe: obiektowe bazy danych, symulacja zdarzeniowa, model obsługi ządań dostępu

Simulation method for analyzing the process of handling data access requests in OODB

Summary

This article presents the simulation method for the efficiency evaluation of the serving access-requests process in object databases. Simulation model has been proposed, capable to investigate the impact of the clustering method on the efficiency of this process. The presented approach assumes organisation of data in the „cache memory-main memory” model. Simulation experiment may be used for a comparison of chosen

clustering methods, at the early stage of designing the final solution. The results can be considered while making decisions about implementation of methods in the real database systems.

Keywords: object-oriented database, discrete-event simulation, model of the serving access-requests

