

Wiesław GRABOŃ

Politechnika Rzeszowska, Zakład Informatyki

TECHNOLOGIE SIECIOWYCH SYSTEMÓW INFORMACYJNYCH Z BAZĄ DANYCH ORACLE

Streszczenie. W artykule przedstawiono sposoby implementacji sieciowych systemów informacyjnych. Opisane zostały mechanizmy działania systemów oraz wynikające z nich wady i zalety.

Słowa kluczowe: Oracle, aplikacje internetowe, PL/SQL, SQL.

TECHNOLOGIES FOR DATA NETWORK SYSTEMS WITH THE ORACLE DATABASE

Summary. The article presents the methods of implementation of data network systems. It describes the mechanisms of system operation as well as advantages and disadvantages resulting from them.

Keywords: Oracle, Web-based application, PL/SQL, SQL.

1. Wstęp

Wszędzie tam, gdzie systemy informatyczne odgrywają dużą rolę, zachodzi potrzeba ich ciągłego udoskonalania. Nowe technologie umożliwiają tworzenie systemów osiągających wyższą wydajność oraz zapewniających większe bezpieczeństwo dla danych.

Wygodnym rozwiązaniem systemu informatycznego jest utworzenie go jako sieciowej aplikacji internetowej. Jedną z technologii projektowania takich aplikacji jest stosowanie rozszerzenia serwera WWW typu CGI [1]. Aplikacja taka instalowana jest na komputerze, na którym pracuje serwer WWW i za pomocą *mechanizmu ODBC* oraz *Oracle Net8* komunikuje się z bazą danych znajdującą się na serwerze *Oracle8*. Podstawowym zadaniem aplikacji jest generowanie dynamicznych stron WWW zawierających dane pochodzące z bazy danych.

Główną wadą aplikacji wykonanej w technologii CGI jest mała wydajność oraz to iż wymaga on znacznych zasobów systemowych ze strony serwera, na którym jest uruchamiana. Gdy aplikacja nie działała bezpośrednio na komputerze, na którym zainstalowany jest serwer *Oracle*, lecz na komputerze, który połączony jest z nim w sieci, czas oczekiwania użytkownika na uzyskanie wyników działania aplikacji znacznie się wydłuża. Wpływ na to mają zarówno mechanizmy, za pomocą których aplikacja łączy się z bazą danych, jak również przepustowość samej sieci.

Do tworzenia aplikacji internetowych współdziałających z bazą danych nowszej generacji *Oracle9i* dostępnych jest kilka technologii [2]. Główną platformą wykonaną przez firmę *Oracle* do uruchamiania aplikacji tworzonych za pomocą niektórych z tych technologii jest *Oracle9i Application Server*. Technologiami najbardziej wspieranymi przez firmę *Oracle* zarówno w zakresie narzędzi programistycznych, jak i obsługi tworzonych w nich aplikacji przez *Oracle9i Application Server* są technologie oparte na języku Java (serwlety oraz PSP) oraz na procedurach składowanych PL/SQL. Istnieją pewne trudności współpracy *Oracle9i* z technologią *PHP*, w której przypadku do *Oracle9i Application Server* należy dodać odpowiedni dodatkowy moduł. Alternatywnym rozwiązaniem dla *Oracle9i Application Server* jest *Oracle HTTP Server* dostarczany przez firmę *Oracle* wraz z serwerem bazy danych *Oracle9i*, bazujący na technologii serwera *Apache*. *Oracle HTTP Server* posiada możliwość uruchamiania aplikacji stworzonych w wymienionych technologiach, jednak w niektórych przypadkach może nie obsługiwać pewnych mechanizmów przez nie dostarczanych.

Z artykułu [2] VII Konferencji PLOUG wynika, że w przypadku aplikacji generujących strony WWW bez odwołania się do bazy danych rozwiązanie oparte na PL/SQL jest zdecydowanie najwolniejsze spośród wymienionych technologii. Jednak daje ono dobre wyniki w aplikacjach, które w sposób intensywny korzystają z danych pochodzących z bazy danych *Oracle*.

2. Koncepcja systemu

W przedstawianej wersji projektowania sieciowego systemu informacyjnego za zadania realizowane przez aplikację wykonaną jako rozszerzenie serwera WWW typu CGI odpowiedzialne są procedury napisane w języku PL/SQL. Działanie w bezpośrednim kontakcie z danymi bazy danych pozwala tym procedurom na uzyskanie wydajnego dostępu do tych danych. Dostęp realizowany jest poprzez zmienne tworzone w pamięci operacyjnej. Polepszenie wydajności uzyskuje się dzięki temu, że zmienne przechowują wartości w tym samym formacie wewnętrznym co system bazy danych. Dlatego np. w momencie pobierania wartości liczbowej z bazy danych nie jest wymagana konwersja – wystarczy bezpośrednie skopiowanie binarnej

postaci danych. Ponadto procedury utworzone w języku PL/SQL są wykonywane po stronie serwera w tym samym programie (w tej samej przestrzeni pamięci) co serwer *Oracle*. Umożliwia to wydajniejszą pracę systemu, gdyż narzut czasowy spowodowany koniecznością komunikowania się z bazą danych jest zmniejszony do minimum [3].

W omawianej wersji przykładowego systemu dotyczącego ewidencji wyników nauczania przez sieć Internet jako serwer WWW wykorzystywany jest *Oracle HTTP Server* dostarczany wraz z serwerem bazy danych *Oracle9i*. Struktura bazy danych, typy użytkowników systemu oraz struktura witryn internetowych, za pomocą których użytkownicy mają dostęp do danych znajdujących się w systemie, pozostała niezmieniona w stosunku do systemu opisanego w artykule [1].

3. Mechanizm działania systemu

Przy współpracy z bazą danych *Oracle9i* system wykonany może być w oparciu o procedury magazynowe PL/SQL, które po uruchomieniu przez *Oracle HTTP Server* generują dynamiczne strony WWW. Cały ten proces przebiega w następujący sposób:

- użytkownik wysyła żądanie do *Oracle HTTPS Server* o daną stronę WWW,
- serwer na podstawie adresu URL określa, że jest to strona generowana przez procedurę PL/SQL i przesyła żądanie do systemu *Oracle*,
- system *Oracle* uruchamia program PL/SQL, który generuje stronę WWW z danymi i przekazuje je do *Oracle HTTP Server*, skąd przesyłana zostaje do przeglądarki w komputerze użytkownika.

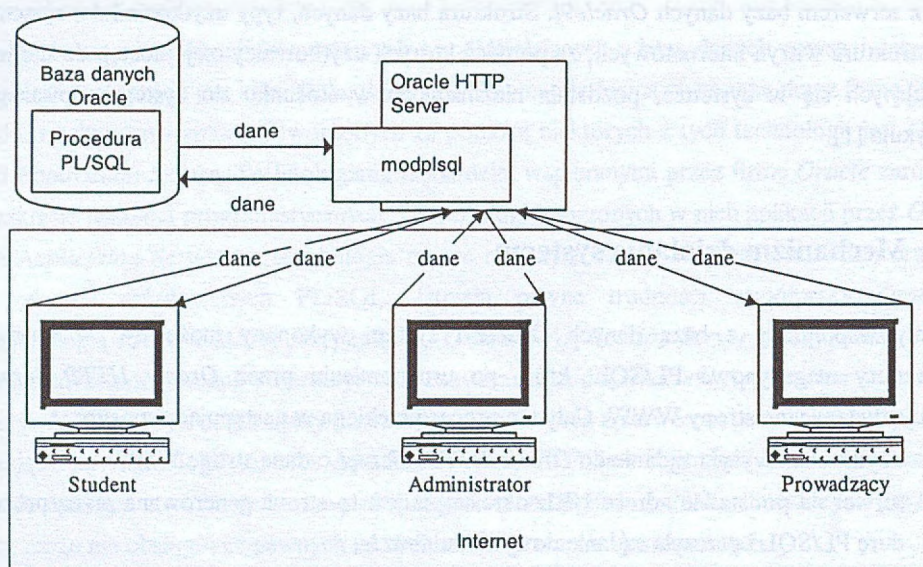
Połączenie pomiędzy serwerem WWW (*Oracle HTTP Server*) a PL/SQL realizowane jest przez program o nazwie *modplsqr*. Użytkownik nie wywołuje tegoż programu bezpośrednio z poziomu kodu procedury, odwołuje się jedynie do wewnętrznych pakietów systemu *Oracle*, które wywołują *modplsqr* zamiast użytkownika. Wywołanie z poziomu kodu PL/SQL jednej z odpowiednich procedur powoduje przesłanie kodu HTML przez *modplsqr* do *Oracle HTTP Server*. Rysunek 1 przedstawia ilustrację mechanizmu działania systemu.

4. Etapy projektowania systemu

4.1. Przeniesienie bazy danych z Oracle8i na Oracle9i

W systemie opisanym w artykule [1] dane przechowywane były w bazie danych *Oracle8*. Do przeniesienia tabel z bazy danych *Oracle8* na *Oracle9i* użyty został program *Export*

i *Import*. Zadaniem tych programów jest obsługa pliku danych o strukturze właściwej dla środowiska *Oracle*. Program *Export* pozwala na wyeksportowanie całej bazy danych, całego schematu danego użytkownika lub pojedynczej tabeli. Ma on możliwość utworzenia jedynie pojedynczego pliku eksportu, co może stanowić pewne ograniczenie przy pracy z dużymi zbiorami danych. Program *Import* może pobierać dane jedynie z plików wygenerowanych przez *Export*, a następnie przenosić je do środowiska *Oracle* [3].



Rys. 1. Mechanizm działania systemu

Fig. 1. Mechanism of system operation

4.2. Tworzenie deskryptora dostępu do bazy danych

Deskryptor dostępu do bazy danych DAD (ang. Database Access Descriptor) zawiera zbiór parametrów konfiguracyjnych używanych przez *Oracle HTTP Server* przy łączeniu się z bazą danych. Parametrami tymi są:

- nazwa połączenia z bazą danych (alias),
- nazwa deskryptora,
- nazwa użytkownika bazy danych,
- hasło użytkownika bazy danych.

Informacje o nazwie i hasle użytkownika są opcjonalne. Jeśli nie zostaną określone w deskrytorze, użytkownik będzie musiał je podać przed wykonaniem logowania do bazy danych. Utworzenie deskryptora umożliwia użytkownikowi, dla którego został on wykonany, wywoływanie poprzez serwer HTTP systemu *Oracle* procedur generujących dynamiczne strony WWW.

4.3. Pobieranie danych z bazy danych do programu PL/SQL

Największy problem z pobieraniem danych wynika z faktu, że SQL jest językiem zorientowanym na obsługę zbioru rekordów, natomiast PL/SQL jest zorientowany na obsługę pojedynczych rekordów. Pobieranie danych staje się więc zadaniem problematycznym, ponieważ wewnętrzne struktury i operatory obydwóch języków nie są do końca zgodne.

Jednym ze sposobów pobierania danych do programu PL/SQL jest użycie do tego celu kursora, który jest nazwą struktury przechowywanej w pamięci, zwanej prywatnym zakresem SQL, przydzielanej przez serwer podczas wykonania każdej instrukcji SQL. Cursor umożliwia pobranie rekordów z bazy danych poprzez operowanie za każdym razem na jednym rekordzie.

System *Oracle9i* oferuje jeszcze inne struktury językowe, zwane kolekcjami (ang. collections) oraz wiązaniami masowymi, które umożliwiają pobieranie wielu rekordów na raz. Dzięki zastosowaniu masowych wiązań istnieje możliwość przechowywania całych zbiorów wyników w lokalnych zmiennych. Wykorzystując kolekcje, będące zmiennymi przechowującymi całe grupy elementów zamiast pojedynczych wartości skalarnych, masowe wiązania umożliwiają przesłanie dużych ilości danych do programu w jednym etapie. Dzięki temu program może je przeglądać w lokalnych zmiennych kolekcyjnych zamiast wykonywać dodatkowe zapytania do serwera [3]. Zastosowanie masowych wiązań daje duże zwiększenie wydajności działania aplikacji. Udokumentowaniem tego są wewnętrzne testy *Oracle* dotyczące masowego wiązania, które wykazały 30% poprawę wydajności. Masowe wiązanie poprawia wydajność od momentu, gdy w programie PL/SQL bez masowych powiązań znajduje się cztery lub więcej iteracji [5].

Zastosowanie masowych wiązań w omawianym systemie pozwoliło na dodatkowe zwiększenie jego wydajności. Usytuowanie systemu na serwerze *Oracle9i* pozwoliło na wykorzystanie jego dodatkowej możliwości, czyli użycia masowych wiązań w wewnętrznym dynamicznym SQL.

Podczas tworzenia systemu znaczące jest również zwrócenie uwagi na fakt, że w trakcie pobierania danych do programu serwer *Oracle* może zwiększyć wydajność wykonywanych instrukcji SELECT dzięki wykorzystaniu współdzielonego SQL (ang. shared SQL). Ważne jest, aby aplikacja była tworzona tak, by w miarę możliwości korzystała z takich samych poleceń do dostępu do tych samych danych. Taki sposób postępowania sprawia, że obszary współdzielone SQL są dobrze wykorzystane. Gdy polecenie jest jak najczęściej dostępne w obszarze współdzielonym, wówczas unika się fazy analizy składni takiego polecenia. Funkcja ta daje największe korzyści w przypadku zastosowania w środowisku wieloużytkownikowym, w którym wiele osób pracuje na jednej bazie danych. Wymagania dotyczące pamięci serwera i mocy jednostki centralnej są tym mniejsze, im mniej indywidualnych instrukcji SQL aplikacja

musi przesłać. Instrukcje SQL muszą być identyczne także pod względem wielkości liter i zastosowanych znaków odstępu [4].

4.4. Tworzenie stron internetowych za pomocą języka PL/SQL

W wersji CGI systemu użytkownik ze stron WWW mógł wykonywać szereg operacji (usunięcie, dodanie lub modyfikacja rekordu w bazie danych). Za wykonanie tych zadań oraz generowanie dynamicznych stron WWW z informacjami z bazy danych odpowiedzialna była aplikacja CGI. W wersji PL/SQL systemu za obsługę wszystkich tych zadań oraz generowanie dynamicznych stron WWW odpowiedzialne są procedury magazynowe napisane w języku PL/SQL.

Tworzenie dynamicznych stron WWW za pomocą języka PL/SQL można wykonywać na dwa sposoby: wstawiając kod HTML do PL/SQL lub na odwrót.

4.4.1. Wstawianie kodu HTML do PL/SQL

W celu wstawienia kodu HTML do PL/SQL można utworzyć zmienną tekstową przechowującą ten kod, a następnie umieścić ją jako parametr w odpowiedniej wbudowanej procedurze PL/SQL, której zadaniem będzie przesłanie go poprzez *modplsqli* do przeglądarki. W projekcie wykorzystano głównie wbudowane procedury HTP.PRINT oraz HTP.PRN.

W procedurach tworzących strony WWW do generacji poszczególnych znaczników HTML można użyć dodatkowych procedur bibliotecznych (np.: *htp.htmlopen*, *htp.htmlclose*). Procedury te mogą być jednak w każdym przypadku zastąpione przez procedurę HTP.PRINT.

Wydruk 1. Procedura przedstawiająca kod HTML wstawiony do PL/SQL

```
CREATE OR REPLACE PROCEDURE dane_pracownicy as
  Type lista_t is table of pracownicy.nazwisko%TYPE;
  Lista lista_t;
  CURSOR lista_kursor IS
  select nazwisko from pracownicy;
BEGIN
  OPEN lista_kursor;
  FETCH lista_kursor BULK COLLECT INTO lista;
  CLOSE lista_kursor;
  htp.prn('<HTML><HEAD><TITLE>Pracownicy</TITLE></Head>' ||
  '<body><table border = "4"><tr><th>Nazwisko</th></tr>')
  for I in 1 .. lista.COUNT
  loop
    htp.prn('<tr><td>' || lista(i) || '</td></tr>' );
  end loop;
  htp.prn('</table></body></html>');
END;
```


W procedurze do zadeklarowanej kolekcji *lista* przekazano w jednej instrukcji cały zbiór rekordów zwracanych przez zapytanie SQL. Taki sposób postępowania nazywany jest masowym wiązaniem. Do obsługi tego wiązania wykorzystano rozszerzenie BULK COLLECT. Za przesłanie kodu *html* do przeglądarki odpowiedzialna jest procedura *htp.prn*, w której jako argument został podany tekst generowanej strony WWW. Do prezentowania zwracanych przez zapytanie SQL wyników na stronie WWW wykorzystany został w omawianej procedurze element *TABLE* języka HTML. Strona generowana przez tę procedurę jest jedną ze stron, do których dostęp ma administrator systemu. W celu łatwiejszej prezentacji procedura w wydruku 1 została przedstawiona w uproszczonej wersji.

4.4.2. Wstawianie kodu PL/SQL do HTML

Tworzenie procedur składowanych generujących strony WWW metodą wstawiania kodu PL/SQL do HTML opiera się na uprzednim przygotowaniu ich w postaci dokumentów *PL/SQL Server Pages* (PSP). Dokument PSP ma postać dokumentu HTML, w którym wewnątrz specjalnych ograniczników zamieszcza się instrukcje dla serwera. Instrukcje te generują dynamiczne części dokumentu. Dokument PSP musi zostać załadowany do bazy danych za pomocą programu *loadpsp*. Program ten odczytuje treść pliku PSP i przekształca ją do postaci procedury magazynowej PL/SQL. Przykładowy dokument przedstawiono na wydruku 2. Dokument ten generuje stronę podobną do strony, którą generuje kod procedury znajdującej się na wydruku 1. W tym przypadku jednak do pobierania danych użyty został kursor o nazwie *kur*.

Wydruk 2. Dokument PSP przedstawiający kod PL/SQL wstawiony do kodu HTML

```
<%@ page language="PL/SQL" %>
<%@ plsql procedure="dane_studenci" %>
<HTML>
<HEAD><TITLE>Studenci</TITLE></HEAD>
<body><table border = "4">
<TR><TH>Nazwisko</TH><TH>Imię</TH>
</TR>
<% for kur in (select nazwisko, imię from studenci) loop %>
  <TR>
    <TD><%= kur.nazwisko %></TD><TD><%= kur.imię %></TD>
  </TR>
<% end loop; %>
</TABLE>
</body>
</HTML>
```

4.5. Obsługa danych pochodzących z formularza

Do przekazywania danych do bazy oraz pobierania od użytkownika kryteriów wyszukiwania podanych na stronach WWW służy zamieszczany na tych stronach mechanizm zwany formularzem *html*. W formularzu takim znajduje się znacznik `<form>`, dla którego jednym z atrybutów jest *action*. Atrybut ten określa miejsce przeznaczenia, do którego dotrą wprowadzone przez użytkownika dane. Wartość tego atrybutu zawiera nazwę procedury PL/SQL odpowiedzialnej za wykonanie założonego przez programistę zadania obsługi wprowadzonych danych.

W momencie przesyłania zawartości formularza program *modplsqli* przekazuje wpisane w każdym elemencie formularza dane jako parametry do procedury odpowiedzialnej za wykonanie danego zadania. Nazwy pól elementów formularza, z których dane przekazywane są do procedury, muszą być takie same jak nazwy parametrów, przez które dane te są do niej wprowadzane.

5. Podsumowanie

Patrząc od strony użytkownika, głównym kryterium oceny efektywności działania systemu jest szybkość jego działania. W przypadku systemów opartych na produktach *Oracle* gwarantuje ją posiadanie dobrze dostrojonego i zoptymalizowanego serwera RDBMS i prawidłowo wykonanej aplikacji. Ważnym elementem potrzebnym do uzyskania wydajnej aplikacji jest również optymalizacja poleceń SQL.

Polepszenie szybkości działania systemów wykonanych w oparciu o programy napisane w języku PL/SQL uzyskuje się również dzięki zwiększeniu wydajności operacji pobierania danych z bazy danych do tych programów. Istnieje wiele sposobów zwiększania wydajności operacji pobierania danych z bazy danych do programu PL/SQL. W omawianym rozwiązaniu największe zastosowanie znalazły masowe wiązania.

Zastosowanie wersji systemu *Oracle9i* pozwoliło również na wykorzystanie dodatkowego atutu umożliwiającego zwiększenie wydajności systemu. Jest nim kompilacja wewnętrzna. Pozwala ona na zwiększenie wydajności działania programów. Dzieje się tak, gdyż podczas kompilacji następuje translacja kodu z PL/SQL do języka C. Następnie system *Oracle* kompiluje kod C do kodu komputera i przechowuje go w bibliotece współdzielonej (ang. shared library), która w czasie wykonywania programu jest wykorzystywana zamiast m-kodu PL/SQL używanego w wersji systemu *Oracle8* [3].

Przeprowadzona za pomocą narzędzia *OpenLoad* symulacja działania systemu, w trakcie której 5 użytkowników równolegle zgłaszało do serwera WWW żądanie o jedną ze stron gen-

erowanych przez obie wersje systemów, dała potwierdzenie lepszej efektywności nowej wersji.

LITERATURA

1. Graboń W.: Obiektowe projektowanie sieciowego systemu informacyjnego. 23 Międzynarodowe Sympozjum Naukowe Studentów i Młodych Pracowników Nauki, Tom: Informatyka, Zielona Góra 2002, ss. 100-107.
2. Wojciechowski M., Zakrzewicz M.: Analiza porównawcza technologii tworzenia aplikacji internetowych dla baz danych Oracle. VIII Konferencja PLOUG, Kościelisko, Październik 2002.
3. Pribyl B., Feuerstein S.: Oracle PL/SQL. Wprowadzenie. Helion, Wrzesień 2002.
4. Greene J. : Oracle8 Server. Księga eksperta. Helion. Gliwice 2000.
5. Sangam U., Swaminathan A., Fogg Oracle8i PL/SQL - szybciej, łatwiej i lepiej. Software 2.0 numer 12/1999.
6. Austin D. : Poznaj Oracle 8. Mikom, Warszawa 1999.
7. Kołodziej P. Czego mądrzy ludzie z Oracle'm nie robią. VI Seminarium PLOUG. Warszawa. Styczeń 2003.
8. Pribyl B., Feuerstein S., Dawes C.: Oracle PL/SQL Language Pocket Reference, O'REILLY.
9. Pipeline Newsletter. <http://www.quest-pipelines.com/newsletter-v2/plsql9i.htm>
10. <http://www.znow.com/sales/oracle/appdev.816/a77069/toc.htm>

Recenzent: Dr inż. Marcin Skowronek

Wpłynęło do Redakcji 6 maja 2003 r.

Abstract

This paper discusses methods for designing the network database systems, as well as the advantages and disadvantages resulting from their design and technologies used for their creation. It also presents technologies that are, to a different extent, supported by the Oracle company in work with its new product *Oracle9i*.

Methods for the network system design with the Oracle9i database are described. This system is shown in Fig. 1. The paper also presents methods for the creation of websites with the use of stored procedures PL/SQL, which is exemplified by printed programs no. 1 and no. 2. The printed program no. 1 shows how to create dynamic websites inserting the HTML code into the PL/SQL. The printed program no. 2 is an exemplary *PL/SQL Server Pages* document, which is the second method for the creation of the discussed websites.

Tools used to import tables from Oracle8 to Oracle9i database are presented. Moreover, the paper discusses role of Database Access Descriptor for Oracle9i, the method for data acquisition from the database for PL/SQL programs, as well as transformation of the data from the website forms with the use of these programs.

What is more, new possibilities offered by Oracle9i, namely the internal compilation and the possibility of the usage of bulk binds in internal dynamic SQL are briefly presented.

Adres

Wiesław GRABOŃ: Politechnika Rzeszowska, Zakład Informatyki, Al. Powstańców Warszawy 8, 35 - 959 Rzeszów, Polska, wgrabon@prz.rzeszow.pl.