

Dorota PIERZCHAŁA

Politechnika Śląska, Instytut Informatyki

REALIZACJA DOSTĘPU DO SQL-OWYCH BAZ DANYCH Z POZIOMU SYSTEMU WWW¹

Streszczenie. W artykule zaprezentowano możliwość wykorzystania serwera baz danych jako źródła informacji dla systemu WWW. Dokonano klasyfikacji funkcjonalnej istniejących rozwiązań. Przedstawiono możliwość rozszerzenia oprogramowania PHP/FI o dostęp do wybranego serwera baz danych. Porównano wykorzystanie SQL-owej bazy danych jako źródła informacji dla WWW z tradycyjnymi, statycznymi plikami na przykładzie pakietu PHP/FI rozszerzonego o możliwość dostępu do serwera SQLBase.

MAKING POSSIBLE THE ACCESS TO SQL-DATABASES FROM WWW SYSTEM

Summary. The possibility of using database server as the source of information for WWW system is presented in the paper. Functional classification of existing solutions is made. The possibility of extending PHP/FI programm to access choosen database server is indicated. The comparison of using SQL-database as the source of information for WWW with traditional, static files, by example of PHP/FI package extended to access SQLBase server is made.

¹Opracowanie wykonano w ramach grantu KBN 8 T11C 020 11.

1. Wprowadzenie

W globalnej sieci komputerowej Internet usługi WWW (World Wide Web) są jednym z najbardziej rozpowszechnionych sposobów udostępniania informacji. Udostępniane za pośrednictwem serwerów WWW informacje to najczęściej prezentacje lokalnych zasobów, dane na temat prezentujących je instytucji lub osób, tematycznie zebrane informacje z różnych dziedzin, publikacje itp.

Prezentowane w systemie WWW informacje przechowywane są zwykle w formie statycznych dokumentów hipertekstowych, bazujących na prostych plikach tekstowych. Rozwiązanie takie ma wiele mankamentów, zwłaszcza gdy ilość prezentowanych informacji stale się zwiększa, jak to występuje obecnie. Użytkownik systemu WWW ma wprowadzić dostęp do coraz większej ilości danych, ale wyszukiwanie istotnych dla niego informacji jest coraz bardziej uciążliwe, gdyż sprowadza się do prostego wyszukiwania, ograniczającego się do znalezienia odpowiednich fragmentów tekstu i nie zmniejszającego przesyłów w sieci.

Podobne problemy w skali sieci lokalnych zostały rozwiązane i w nich standardem gromadzenia i prezentowania informacji są relacyjne bazy danych, najczęściej oparte na architekturze klient-serwer i języku zapytań SQL. Ten sposób przechowywania danych ułatwia nie tylko ich przeszukiwanie, nawet według złożonych kryteriów, oraz aktualizację, ale również pozwala na wyeliminowanie ich redundancji oraz niespójności.

Udogodnienia te inspirują do połączenia zalet standardowego, graficznego interfejsu użytkownika charakterystycznego dla WWW, z zaletami serwerów SQL jako źródła danych prezentowanych w globalnej sieci komputerowej (zwłaszcza że prezentowane na serwerach WWW informacje są często kopią danych przechowywanych w bazach danych na serwerach lokalnych).

Efektem badań jest zestaw narzędzi przydatnych programiście do łatwego tworzenia dostępu do wybranej bazy danych z poziomu systemu WWW oraz przykładowe wykorzystanie tych narzędzi, umożliwiające dostęp do baz danych na serwerze SQLBase. Porównano również wykorzystanie bazy danych jako źródła informacji dla WWW z tradycyjnymi, statycznymi plikami.

2. Przegląd i klasyfikacja istniejących rozwiązań

W ciągu trwania badań zebrano informacje na temat wielu rozwiązań problemu publikowania informacji pochodzących z relacyjnych baz danych na stronach WWW (64 programy dla systemu Unix, w tym 36 freeware'owych i shareware'owych - najczęściej uniwersyteckich, oraz 62 przeznaczone dla komputerów PC z systemami DOS i Windows).

Najstarsze z tych narzędzi ukazały się z końcem 1995 roku i nie umożliwiają pełnej integracji baz danych z serwerem WWW.

W niniejszym opracowaniu nie zostaną zamieszczone szczegółowe informacje na temat poszczególnych narzędzi. Celowe natomiast wydaje się ogólne podanie charakterystycznych cech i najważniejszych różnic funkcjonalnych, które pozwalają podzielić istniejące tego typu programy na pewne klasy rozwiązań.

2.1. Kryteria klasyfikacji istniejących rozwiązań

• Baza danych:

- Liczba wspieranych baz danych:
 - jedna, dedykowana,
 - wiele.
- Zakres dostępnych baz danych:
 - rozszerzalny,
 - nierozszerzalny.
- Rozwiązania oryginalne:
 - np. serwer WWW z wbudowaną, specjalnie zaprojektowaną bazą danych (WWWData).

Część dostępnych narzędzi pochodzi wprost od producentów serwerów baz danych (np. Oracle WebServer, Informix WebKit) i w tym przypadku istnieje możliwość wykorzystania danych tylko z jednej, określonej bazy danych. Istnieje jednak klasa narzędzi umożliwiających dostęp do wielu różnych baz danych. Narzędzia te można podzielić na takie, które komunikację z bazą danych realizują za pomocą ODBC (Open Database Connectivity) (z wszystkimi tego rozwiązania wadami i zaletami) - np. Microsoft IIS+IDBC oraz takie, które wykorzystują dla każdej bazy danych, charakterystyczny dla niej, naturalny interfejs (najczęściej w postaci interfejsu API), jak Centura, czy PHP/FI.

Większość narzędzi komercyjnych ma nierozszerzalny zakres dostępnych baz danych (jedna lub więcej, ale ściśle określonych), natomiast oprogramowanie dostarczane darmowo, zwłaszcza w postaci kodu źródłowego, można zwykle samodzielnie rozszerzyć o interfejs do dowolnie wybranej bazy danych.

Osobną kategorię pod względem udostępnionych baz danych tworzą narzędzia w postaci serwera WWW z wbudowaną, specjalnie zaprojektowaną do tego celu, bazą danych, czego przykładem jest WWWData.

Rodzaj wspieranych zapytań SQL:

- tylko selekcja danych,
- możliwość modyfikacji danych.

Pierwsze z tworzonych aplikacji stanowiły wyłącznie narzędzia ułatwiające i wspomagające wyszukiwanie danych i umożliwiały wyłącznie selekcję informacji z bazy danych, później pojawił się problem np. rejestracji użytkowników, profesjonalnej i bezpiecznej realizacji zakupów w Internecie czy wykorzystywania podobnych narzędzi w sieciach intranetowych, które wymagały istnienia programów umożliwiających wykonywanie modyfikacji danych przez użytkowników sieci.

Połączenie do bazy danych:

- Nawiązywanie połączenia:
 - nawiązywane wielokrotnie (na każdej stronie),
 - trwałe.
- Autoryzacja:
 - wszyscy klienci WWW pracują jako jeden określony użytkownik,
 - normalna autoryzacja, jak w lokalnej sieci.

Najłatwiej i najefektywniej jest zrealizować połączenie do bazy danych na czas generacji pojedynczej, dynamicznej strony WWW. Wynika to przede wszystkim z zasad działania protokołu HTTP (Hyper Text Transfer Protocol) i z faktu, że w odróżnieniu od działania tradycyjnych aplikacji baz danych w architekturze klient-serwer, użytkownik Internetu może przełączać się między stronami WWW w dowolnej, nieprzewidywalnej kolejności. Rolą oprogramowania jest natomiast zapewnienie istnienia połączenia do bazy danych przed wykonaniem każdej innej operacji z nią związanej oraz z drugiej strony sprawienie, aby połączenia te nie były zbyt długotrwałe, ze względu na limitowaną zwykle liczbę jednoczesnych połączeń do bazy danych i blokowanie zasobów.

Ponieważ połączenia i rozłączenia z bazą danych są dość kosztowne czasowo, skonstruowano również narzędzia, które zapamiętują połączenie do bazy danych pomiędzy stronami.

Innym problemem jest autoryzacja użytkowników. Często istnieje konieczność zróżnicowania uprawnień użytkowników, tak jak to występuje dla użytkowników baz danych w tradycyjnych rozwiązaniach lokalnych. Nie wszystkie programy mają takie możliwości. Duża ich część umożliwia pracę wszystkim użytkownikom Internetu z uprawnieniami jednego, określonego użytkownika bazy danych. Klasyczna autoryzacja użytkowników jest łatwiejsza

do zrealizowania w systemach zarządzania bazami danych, mających własny system użytkowników, niezależny od systemu operacyjnego lub go uzupełniający.

Przechowywanie stron WWW:

- tradycyjnie, w plikach,
- na serwerze baz danych.

Zdecydowana większość narzędzi przechowuje szablony dynamicznych stron WWW lub strony wygenerowane na podstawie zawartości baz danych w sposób tradycyjny, czyli w tekstowych plikach ASCII. Pojawiają się jednak oryginalne rozwiązania (np. Sybernet), które są tak ścisłym powiązaniem serwera baz danych i serwera WWW, że nawet strony WWW przechowywane są w bazie danych (w tym przypadku Sybase).

Aktualizacja stron dynamicznych:

- w wyniku zapytania,
- w wyniku modyfikacji bazy danych (z wykorzystaniem procedur bazy danych),
- co określony czas.

Sposób i częstotliwość aktualizacji danych prezentowanych na stronach internetowych, a będących odzwierciedleniem zawartości bazy danych, powinny być ściśle uzależnione od zastosowania: częstości tychże zmian w samej bazie danych, jak i tego, czy istnieje konieczność prezentowania danych jak najbardziej aktualnych. W systemach, które tworzą strony dynamiczne "w locie", w wyniku zażądania prezentacji tych stron przez klienta WWW, dane są pobierane z bazy danych dla każdego takiego żądania i w momencie tworzenia strony są to na pewno dane aktualne. Jeżeli dane w bazie są rzadko aktualizowane i dane prezentowane różnym użytkownikom są te same, a klientów WWW odwołujących się do stron HTML (Hyper Text Markup Language) jest dużo, efektywniejszym rozwiązaniem może być przechowywanie gotowych stron generowanych na podstawie bazy danych co określony przedział czasu. Jeśli trudno jest dobrać ten odcinek czasu i zależy nam na prezentacji danych aktualnych po każdej ich modyfikacji (gdy modyfikacje nie są zbyt częste, a prezentowane dane są powtarzalne) interesującym rozwiązaniem jest uzupełnienie aktywnej bazy danych o procedury, które przy każdej modyfikacji przydatnych nam danych będą aktualizowały odpowiednie strony WWW.

Sposób integracji z serwerem WWW:

- Liczba wspieranych serwerów:
 - przeznaczone do współpracy z określonym serwerem,
 - współpracujące z wieloma serwerami.
- Sposób połączenia z serwerem WWW:
 - dołączana biblioteka programów,
 - moduł dołączany do serwera.

Obsługa interfejsu użytkownika:

- CGI,
- Java,
- inne (np. ISAPI - Internet Server Application Programmer Interface, często niewidoczne dla programisty).

Sposób definiowania dynamicznych stron WWW:

- rozszerzenia języka HTML,
- pliki konfiguracyjne,
- na podstawie struktury bazy danych,
- inne.

Sposoby definiowania stron opartych na informacjach pochodzących z baz danych są bardzo zróżnicowane wśród analizowanych programów. Najsilniejszym narzędziem dla programisty są z pewnością rozszerzenia języka HTML o nowe funkcje i konstrukcje programowe; niektóre z programów tworzą wręcz własny, funkcjonalnie pełny język programowania, którego częścią składową jest możliwość wypisania znaczników HTML, tworzących stronę WWW. Wadą tego rozwiązania, szczególnie dla twórców stron WWW nie będących informatykami, jest konieczność opanowania dodatkowego języka programowania. Prostszy, lecz mniej elastycznym rozwiązaniem jest oparcie konstrukcji stron na plikach konfiguracyjnych, zawierających np. informacje na temat połączenia poszczególnych obiektów ekranowych z obiektami w bazie danych oraz brzmienie pytania w języku SQL, którego wynik ma zostać zaprezentowany.

Istnieją także narzędzia, które tworzą strony internetowe bezpośrednio na podstawie struktury bazy danych, ale ten sposób nadaje się raczej do prezentacji wszystkich informacji z bazy danych, najczęściej specjalnie do tego celu zaprojektowanej i wykorzystywanej.

3. Dobór narzędzi do wykonania badań

Oprogramowanie tworzone w ramach badań ze względu na swoje przeznaczenie powinno spełniać następujące kryteria:

1. Umożliwiać zarówno selekcję, jak i modyfikację danych zgromadzonych na wybranym serwerze baz danych.
2. Udostępniać normalną autoryzację użytkowników bazy danych, jak w rozwiązaniach lokalnych.
3. Aktualizować dynamiczne strony WWW w wyniku zapytania pochodzącego z przeglądarki, gdyż zakładamy, że prezentowane informacje zostają poddane dynamicznym warunkom

selekcji nakładanym ze strony użytkownika, przez co stanowić mogą niewielki fragment informacji zgromadzonych w bazie danych, inny dla każdego selekcjonującego użytkownika.

4. Najlepiej - nie być zależnym od określonej architektury sprzętowo-programowej (być w pełni przenaszalnym), w szczególności od konkretnego serwera WWW.

5. Tworzyć rozszerzenie języka HTML w postaci nowych funkcji, dla pełnej elastyczności zastosowań.

Nie jest natomiast istotne:

1. Nawiazywanie połączenia do bazy danych - trwale, czy na każdej stronie.

2. Miejsce przechowywania stron WWW.

3. Sposób integracji z serwerem WWW (biblioteka programów czy dołączany moduł).

4. Sposób obsługi interfejsu użytkownika.

Na podstawie przeprowadzonej analizy dostępnych narzędzi, w celu podniesienia stopnia użyteczności tworzonego oprogramowania, została podjęta decyzja o dołączeniu projektowanej biblioteki programów do kodu źródłowego oprogramowania już istniejącego. Ze względu na ustalone wcześniej kryteria, jako bazowe oprogramowanie został wybrany pakiet PHP/FI.

W ramach badań postanowiono stworzyć interfejs pozwalający na dostęp do baz danych na serwerze SQLBase (firmy Gupta, obecnie Centura), gdyż dane przykładowe gromadzone są w takim właśnie formacie. Wykorzystywanym serwerem World Wide Web jest oprogramowanie Apache w wersji 1.2 pracujące pod kontrolą systemu operacyjnego SunOS 4.1.3, a serwerem baz danych serwer SQLBase 5.0 w wersji dla systemu operacyjnego SunOS.

3.1. Krótka charakterystyka oprogramowania PHP/FI

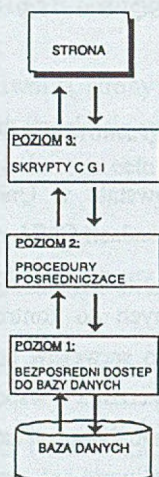
PHP/FI jest to oprogramowanie powstałe w University of California, którego kod źródłowy dostarczany jest za darmo na zasadach GNU, poprzez sieć Internet. Udostępnia ono język skryptów umożliwiający tworzenie dynamicznych stron, rozszerzając możliwości samego języka HTML o wiele funkcji potrzebnych do kontroli interfejsu użytkownika. Zawiera również funkcje umożliwiające dostęp do serwerów baz danych: msql, Postgres95 i Solid (wersja z czerwca 1997 roku została poszerzona o wsparcie dla serwerów Sybase i Oracle). Za pośrednictwem tego oprogramowania możliwa jest zarówno prezentacja, jak i modyfikacja danych przez klienta WWW. Istnieje możliwość autoryzacji użytkowników przy dostępie do poszczególnych stron, i/lub określania domen z odpowiednimi uprawnieniami. PHP/FI jest dostępne w wersji dla różnych systemów Unix'owych oraz od najnowszej wersji - dla Microsoft Windows NT i Windows 95. Jest przeznaczone do współpracy z różnymi serwerami WWW w postaci dynamicznych bibliotek, istnieje też możliwość skompilowania go jako modułu serwera Apache, z którym jest najlepiej zintegrowane.

Strony WWW, będące plikami zawierającymi fragmenty w języku skryptów PHP/FI, są przed wysłaniem do klienta interpretowane przez serwer WWW, który wykonuje wszystkie polecenia i funkcje PHP/FI (m.in. kieruje zapytania do bazy danych), tworząc w locie strony w czystym języku HTML. Istotną zaletą jest możliwość dołączania do PHP/FI własnych funkcji i bibliotek, dzięki czemu możliwe jest na przykład rozszerzenie liczby udostępnianych baz danych.

4. Realizacja dostępu do baz danych na serwerze SQLBase z poziomu systemu WWW

Realizację dostępu do serwera baz danych z poziomu WWW można ogólnie zdekomponować na następujące poziomy (rys. 1):

- bezpośredni dostęp do bazy danych (przesył zapytania do serwera i odbiór odpowiedzi),
- procedury pośredniczące, formułujące zapytania w języku SQL i przygotowujące wynik do prezentacji,
- skrypty *cgi* (ang. common gateway interface), wzbogacające strony HTML o elementy interakcji z użytkownikiem,
- konstrukcja stron w języku HTML.



Rys. 1. Wielopoziomowa struktura projektu

Fig. 1. Multilevel project structure

Poziom 3: programy wywoływane bezpośrednio z dokumentu HTML przez klienta HTTP. Programy te pobierają dane wejściowe (ze standardowego wejścia lub przez zmienne

środowiska). Kiedy dane wejściowe są odczytane (tu najczęściej zapytanie SQL lub jego fragmenty), zostają przekazane poprzez procedury pośredniczące do poziomu 1.

Poziom 2: jest interfejsem charakterystycznym dla konkretnego serwera baz danych, służącym do translacji zapytań do formatu akceptowanego przez ten serwer; zawiera również opis struktur danych, definicje zmiennych i wywołania funkcji potrzebne do komunikacji z bazą danych.

Poziom 1: zwykle jest oprogramowaniem dostarczonym przez producenta bazy danych, pozwala na dostęp do bazy danych przez API lub ODBC.

Dobór narzędzi (serwera baz danych oraz oprogramowania PHP/FI) w dużym stopniu determinuje funkcje realizowane przez poziom 1 - bezpośredni dostęp do bazy danych oraz poziom 3 - skrypty CGI. W opisywanych badaniach wykorzystana została biblioteka SQLBase C Application Programming Interface (C/API) pozwalająca na pisanie aplikacji w języku C, wykorzystujących dane zgromadzone na serwerze SQLBase. Poziom 3 natomiast został zaprojektowany jako skrypty pisane przy użyciu PHP/FI, umożliwiające tworzenie dynamicznych stron HTML. W celu zapewnienia dostępu do baz danych na serwerze SQLBase język skryptów PHP/FI należało poszerzyć o funkcje odwołujące się do tego serwera. Poziom 2 to zestaw funkcji pośredniczących między poziomami pierwszym i trzecim napisanych w języku C z wykorzystaniem SQLBase C/API. Funkcje te pisane są w konwencji umożliwiającej dołączenie ich jako dodatkowego modułu do PHP/FI.

4.1. Rozszerzenie PHP/FI o funkcje realizujące dostęp do serwera SQLBase

Dla wygody i pełnej funkcjonalności przy tworzeniu stron opartych na zawartości bazy danych potrzebna była realizacja następujących zadań:

1. Połączenie do dowolnej bazy danych na udostępnianym serwerze z dostępem autoryzowanym poprzez podanie nazwy użytkownika i hasła.
2. Posiadania wielu jednoczesnych połączeń do bazy danych.
3. Zamykanie dowolnie wybranych połączeń do bazy danych.
4. Wykonywanie następujących operacji na bazie danych:
 - a) selekcji danych (instrukcja SELECT w języku SQL),
 - b) modyfikacji danych (instrukcja UPDATE),
 - c) dopisywania nowych danych (instrukcja INSERT),
 - d) zatwierdzanie lub wycofywanie transakcji (COMMIT i ROLLBACK).
5. Ze względu na przeznaczenie programu - zabezpieczenie, by żadna inna operacja, prócz wymienionych w powyższym punkcie, nie mogła być zrealizowana.
6. Po realizacji zapytań selekcyjnych dane z bazy danych, uzyskiwanie następujących informacji w celu wygodnej prezentacji danych:

- a) liczba wierszy będących odpowiedzią na zadane zapytanie,
- b) liczba kolumn wchodzących w skład powyższego wyniku,
- c) nazwy poszczególnych kolumn (zwłaszcza dla zapytań zadawanych dynamicznie przez użytkowników),
- d) możliwość przesunięcia się na następny wiersz w obrębie kursora,
- e) pobranie danych z odpowiedniej kolumny i wiersza, na który wskazuje kursor.

Funkcje realizujące powyższe operacje napisane zgodnie z przygotowaną specyfikacją (której zamieszczenie wykracza poza ramy niniejszego opracowania) były tworzone z myślą o dołączeniu ich do pakietu PHP/FI. Poszerzenie PHP/FI o własne funkcje wymaga następujących operacji:

• Uzupełnienie gramatyki PHP/FI o definicję dodawanej funkcji

Jeśli dołączana funkcja posiada od 0 do 6 parametrów, nie ma trzeba rozszerzać gramatyki oprogramowania PHP/FI.

Gramatyka PHP/FI jest definiowana w pliku *parse.raw*, wchodzącym w skład plików z kodem źródłowym pakietu. W celu jej uzupełnienia należy dodać do niej atom, będący słowem kluczowym zawierającym zapisaną wielkimi literami nazwę dodawanej funkcji. Definicje atomów znajdują się na początku pliku *parse.raw*, a kolejność ich występowania nie jest istotna. Gramatyka budowana jest zgodnie z regułami programu YACC (Yet Another Compiler Compiler).

Wszystkie funkcje realizujące dostęp do serwera SQLBase zostały zaprojektowane jako funkcje pobierające od jednego do trzech parametrów ze stosu wyrażeń, więc nie było konieczne uzupełnianie gramatyki PHP/FI.

• Dodanie nazwy funkcji do tablicy mieszającej analizatora leksykalnego

Tablica mieszająca znajduje się w pliku *lex.c*, wchodzącym w skład plików z kodem źródłowym PHP/FI. Opis tablicy mieszającej rozpoczyna się wierszem:

```
static cmd_table_t cmd_table[22][30] = {
```

definiującym dwuwymiarową tablicę programową przechowującą tablicę mieszającą. Wartość 22 jest liczbą o jeden większą od maksymalnej długości nazwy funkcji, a 30 odnosi się do liczby funkcji w każdej z list. Jeżeli dodanie własnych funkcji wymaga zwiększenia podanych limitów - należy to uczynić w definicji tablicy mieszającej, zmieniając jej rozmiary.

Tablica mieszająca jest skonstruowana w bardzo prosty sposób - wartość w tablicy mieszającej odpowiada długości łańcucha znaków zawierającego nazwę funkcji. Na przykład dla funkcji *GDisconnect* trzeba było dodać odpowiedni wiersz dla wartości 11 w tablicy mieszającej:

```
{ "GDisconnect", INTFUNC1, GDisconnect },
```

W wierszu tym przypisujemy nazwie dołączanej funkcji znaczenie atomu INTFUNC1, co oznacza jednoargumentową wewnętrzną funkcję (ang. *internal function*). Pierwszy z

łańcuchów znakowych, objęty cudzysłowem, jest nazwą funkcji, do jakiej można odwoływać się z plików HTML, a nazwa bez cudzysłowu powinna odpowiadać oryginalnej nazwie funkcji z jej definicji.

• Napisanie kodu źródłowego dołączanej funkcji

Własne funkcje można tworzyć za pomocą dowolnych języków programowania, o ile zachowują one konwencję wywoływania funkcji charakterystyczną dla języka C. Należy również zapewnić możliwość stworzenia pliku biblioteki (lub obiektów) kompatybilnego z posiadanym programem konsolidującym, w celu połączenia tego pliku z PHP/FI.

Funkcje umożliwiające dostęp do serwera SQLBase zostały napisane w języku C i wszystkie zawarte są w jednym pliku (*gupta.c*). Przy pisaniu dołączanych funkcji należy pamiętać o sposobie przekazywania danych i zwracania przez nie wyników: sama funkcja powinna być zadeklarowana jako bezparametrowa procedura, np:

```
void GDisconnect(void)
```

Argumenty funkcji pobierane są ze stosu wyrażeń i na ten stos odkładany jest wynik jej działania. Stos wyrażeń jest stosem łańcuchów znakowych reprezentujących zmienne różnych typów oraz związanych z nimi typów danych. PHP/FI rozróżnia trzy podstawowe typy zmiennych: STRING (łańcuch znaków), LNUMBER (długa liczba całkowita), DNUMBER (liczba zmiennoprzecinkowa, podwójnej precyzji).

• Dodanie prototypu funkcji

Wszystkie dostępne z poziomu PHP/FI funkcje muszą mieć swoje prototypy w pliku nagłówkowym *php.h*. Prototypy funkcji w tym pliku pogrupowane są według nazw plików, w których występują ich definicje. Plik ten należy odpowiednio uzupełnić o wiersze opisujące dodawane funkcje, np.:

```
void GDisconnect(void);
```

• Kompilacja

Jeżeli wszystkie elementy oprogramowania są już przygotowane, można przystąpić do kompilacji i konsolidacji pakietu PHP/FI wzbogaconego o własne funkcje. W przypadku wcześniejszej modyfikacji pliku *parse.raw*, przy tworzeniu gramatyki, należy najpierw wykonać polecenie *make parser*, po czym następuje zwykła kompilacja poleceniem *make*.

4.2. Konstrukcja dynamicznych stron WWW na podstawie zawartości bazy danych

Wspomniano już, że w przypadku tradycyjnych, statycznych stron WWW, serwer World Wide Web tylko udostępnia klientom dokumenty w języku HTML, które są interpretowane przez przeglądarki po stronie klienta.

PHP/FI udostępnia bogaty język skryptów, który może być zanurzony w języku HTML (i jednocześnie, w którym język HTML może być zanurzony), a którego polecenia są interpretowane przez serwer WWW, przed wysłaniem gotowego dokumentu, przetłumaczonego na "czysty", statyczny HTML do przeglądarki klienta.

W zależności od sposobu instalacji oprogramowania PHP/FI oraz od samego serwera WWW pewne dokumenty są interpretowane przez parser wchodzący w skład pakietu PHP/FI. Fragmenty kodu pisane w języku skryptów udostępnianym przez PHP/FI mogą być zamieszczane w dowolnym miejscu dokumentu HTML-owego, w obrębie znaczników: rozpoczynającego - `<?` i kończącego `>`. W skryptach PHP/FI dostępne są trzy podstawowe typy zmiennych: liczby całkowite i podwójnej precyzji oraz łańcuchy znakowe; można tworzyć tablice tych zmiennych.

Instrukcje języka udostępnianego przez PHP/FI (jest ich 13) są proste, ich składnia zbliżona jest do języka C. Zestaw dostępnych funkcji jest stosunkowo bogaty i zapewnia wygodne środowisko pracy.

Po rozszerzeniu PHP/FI o funkcje realizujące dostęp do serwera SQLBase można w prosty sposób tworzyć dynamiczne strony WWW spełniające następujące zadania:

- umożliwienie formułowania kryteriów wyszukiwania przez użytkownika zgodnie ze standardem Query By Form,
- umożliwienie zadawania dowolnych zapytań w języku SQL,
- prezentacja informacji z baz danych, będących wynikiem zapytania QBF i SQL,
- realizacja kontrolowanej aktualizacji informacji zgromadzonych w bazie danych.

Możliwości te zostały wykorzystane do stworzenia stron WWW prezentujących dane na temat zbiorów biblioteki Instytutu Informatyki. Informacje te przechowywane są na serwerze SQLBase. W bazie danych zgromadzono informacje na temat około 1500 pozycji książkowych.

Poniżej zamieszczono wygląd stron WWW skonstruowanych za pomocą stworzonych w ramach badań narzędzi.

Kryteria wyszukiwania:

Numer inwentarza:

Tytuł:

Autorzy:

Rok wydania:

Język: Wydawca:

[Powrót na początek](#)

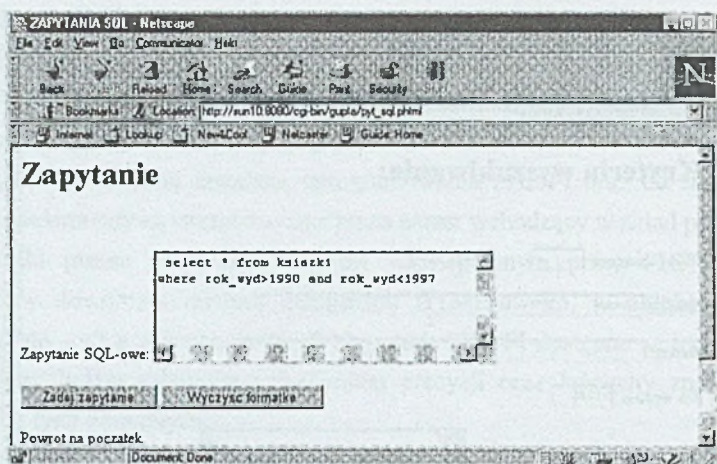
Rys. 2. Przykład strony WWW - kryteria selekcji użytkownika
 Fig. 2. Example of WWW page - user's select criteria

Wierszy wynikowych jest 42

Pierwsze 20 wierszy

nr_inw	tytuł	autorzy	rok_wyd	nazwa_wyd	adres_wyd	język
3127	Entity - relationship		1991	North - Holland Publ. Company	Amsterdam	angielski
3246	Parallel processing		1991	IEEE CS Press		angielski
3451	Deductive and ...		1991	Springer-Verlag	Berlin	angielski
3470	Computer animation '91		1991	Springer-Verlag	Tokyo	angielski
3089	Parallel and Distributed Processing		1992	IEE Computer Society	Los Alamitos	angielski
3119	Combinational		1992	Springer-Verlag	Berlin	angielski
3131	Combinatorics		1992	North - Holland Publ. Company	Amsterdam	angielski
3140	Windows for Workgroups		1992	Microsoft Corporation	Redmond	angielski
3160	Local Area Network		1992	National Semco. Corporation	Santa Clara	angielski
3166	Parallel and Distributed		1992	IEEE CS Press		angielski
3167	Frontiers of Manirely		1992	IEEE CS Press		angielski

Rys. 3. Przykład strony WWW - prezentacja wyniku zapytania
 Fig. 3. Example of WWW page - presenting of query's result



Rys. 4. Przykład strony WWW - zapytania SQL

Fig. 4. Example of WWW page - SQL queries

5. Porównanie wykorzystania SQL-owej bazy danych jako źródła informacji dla WWW z tradycyjnymi statycznymi plikami

Generacja stron World Wide Web bezpośrednio na podstawie zawartości relacyjnej bazy danych niesie ze sobą wiele udogodnień, między innymi można dokładnie precyzować warunki selekcji danych i dowolnie łączyć ze sobą logicznie wiele warunków, w przeciwieństwie do statycznych dokumentów, w których możliwe jest wyłącznie proste przeszukiwanie tekstu. W tradycyjnym rozwiązaniu użytkownik zwykle musi przetransmitować poprzez sieć wszystkie udostępniane informacje, a następnie w tekście poszukiwać interesującego go wzorca. Przy dużych ilościach informacji jest to często uciążliwe i długotrwałe.

Wykorzystanie serwera baz danych ma dwie ważne zalety: warunki selekcji mogą być bardziej precyzyjne (nakładane na poszczególne kolumny, bogate w funkcje i operatory itp.), a przez sieć transmitowany jest tylko wynik wyszukiwania, zwykle będący znacznie mniejszą porcją danych, co zmniejsza czas transmisji i wpływa korzystnie na wygodę przeglądania zbioru wynikowego.

Z drugiej strony należało zbadać, czy dodanie dodatkowej warstwy oprogramowania dla serwera WWW, która interpretuje dynamiczne strony oraz czas przetwarzania zapytań przez serwer baz danych, będzie znacząco wpływać na czas jej pozyskiwania przez przeglądarkę.

Do przeprowadzenia testów porównawczych czasu dostępu do danych zgromadzonych na serwerze baz danych oraz pochodzących z tradycyjnych plików HTML-owych wygenerowana została statyczna strona zawierająca tabelę z informacjami na temat wszystkich książek na

podstawie zawartości bazy danych, o graficznym formacie dokładnie takim samym, jak na dynamicznej stronie prezentującej dane z bazy danych.

Testy wykonywane były z wykorzystaniem tego samego serwera WWW, z jednej stacji klienta i były wykonywane kilkakrotnie na zmianę, by wyeliminować wpływ zmian obciążenia sieci i wydajności komputerów na uzyskiwane wyniki. Nie są tu zamieszczane szczegółowe opisy platformy sprzętowo-programowej przeprowadzonych badań, gdyż nie są istotne pojedyncze wartości osiągniętych wyników, a jedynie uzyskiwane różnice pomiędzy wynikami.

Pierwszym zadaniem testowym było porównanie czasu pozyskiwania statycznego pliku HTML zawierającego dane na temat wszystkich książek (wielkości około 340 kB) oraz pozyskania analogicznej strony wygenerowanej na podstawie zawartości bazy danych. Na stronie dynamicznej zmierzono również programowo czas jej przetwarzania (interpretację przez PHP/FI i czas uzyskiwania informacji z bazy danych SQLBase). Otrzymano następujące średnie wyniki:

Dane pochodzące z pliku	82 s
Dane pochodzące z bazy danych	120 s (w tym przetwarzanie 41 s)

W obu przypadkach czas transmisji danych jest taki sam i zależy od szybkości łącza sieciowego. Dane pochodzące z bazy danych były pozyskiwane w czasie o około 42 s dłuższym, co w tym przypadku stanowi prawie 30% czasu oczekiwania. Wydaje się to dużym narzutem, lecz badania były prowadzone w sieci lokalnej, z szybkością transmisji ok. 10 Mb/s. W przypadku gorszej przepustowości łącza czas pozyskiwania danych może znacznie się wydłużyć, a czas przetwarzania strony dynamicznej nie powinien ulec zmianie przy podobnym obciążeniu serwera baz danych i wówczas wprowadzony narzut będzie stanowił mniejszą część czasu oczekiwania na dane.

Test ten był tylko sprawdzianem pokazującym ewentualne wady wprowadzenia relacyjnej bazy danych jako źródła informacji dla serwera WWW. Bardziej typowym przykładem wykorzystania serwera WWW i dynamicznych stron jest selekcja danych według określonego kryterium i prezentacja informacji zawężonych do wyniku tej selekcji.

W statycznych plikach HTML możliwość wyszukiwania informacji na nich prezentowanych sprowadza się do:

- przetransmitowania i wyświetlenia całej strony,
- wyszukiwania wzorca w tekście tej strony (funkcja przeglądarki),

Natomiast w przypadku rozwiązania wypracowanego w ramach badań selekcja sprowadza się do:

- przesłania zapytania do serwera baz danych przez oprogramowanie pośredniczące,

- wykonania zapytania przez serwer,
- przesyłu zbioru wynikowego z serwera baz danych do oprogramowania pośredniczącego,
- konstrukcji strony zawierającej tylko wynik zapytania,
- przetransmitowania przygotowanej strony do przeglądarki.

Wykonano testy porównawcze czasów trwania powyższych operacji dla obu rozwiązań (statycznego i dynamicznego). W przypadku tradycyjnego dokumentu w języku HTML wyróżniono oba etapy (transmisję i wyszukiwanie), natomiast w przypadku bazy danych jako źródła informacji zmierzono łączny czas trwania operacji uzyskiwania wyniku, aby poprzez wprowadzanie wielu pomiarów na dynamicznych stronach nie zaburzać rzeczywistego czasu trwania tej operacji. Pomiary zostały wykonane dla serwera SQLBase i programów pośredniczących będących efektem poprzednich zadań badawczych oraz dodatkowo dla serwera postrelacyjnej bazy danych Postgres 95, którego obsługa jest integralną częścią PHP/FI. Oba serwery uruchamiane były na tym samym komputerze i udostępniały bazę danych o takiej samej zawartości. Testy dla baz danych podzielone zostały na dwa rodzaje: test alfanumeryczny (wyszukiwanie książek zawierających w dowolnym miejscu w tytule poszukiwany wzorec; w tym przypadku na obu serwerach indeksy nie są używane) oraz numeryczny (wyszukiwanie według dokładnej wartości numerycznej w obrębie poindeksowanego pola). Wyszukiwane wartości były dobrane tak, aby tylko jeden wiersz spełniał warunki selekcji, gdyż to najlepiej daje się porównać z przeszukiwaniem statycznych dokumentów. Poniżej prezentowane są uzyskane rezultaty.

Selekcja alfanumeryczna

Tabela 1

Źródło danych	Średni czas
Serwer SQLBase	1.79 s
Serwer Postgres 95	3.91 s
Statyczny plik HTML	80 s (transmisja)+ 33 s (przeszukiwanie)

Selekcja numeryczna

Tabela 2

Źródło danych	Średni czas
Serwer SQLBase	1.43 s
Serwer Postgres 95	2.75 s
Statyczny plik HTML	80 s (transmisja)+ 38 s (przeszukiwanie)

Wyniki testów pokazują wyraźną przewagę czasową wyszukiwania realizowanego przez serwery baz danych. Nawet w przypadku gdy wynikiem wyszukiwania są większe ilości danych narzut czasowy na przetwarzanie dynamicznej strony nie jest dominujący i nie dyskwalifikuje tego rozwiązania.

Dzięki zastosowanym mechanizmom bezpieczeństwo bazy danych udostępnianej w sieci Internet (w sensie poufności danych) niemalże nie różni się od tradycyjnych rozwiązań lokalnych. Sposób autoryzacji użytkowników jest zrealizowany tak, jak w typowych aplikacjach baz danych w architekturze klient-serwer. W pełni obowiązującym jest system uprawnień użytkowników nadawanych odnośnie do pojedynczych obiektów w bazie danych. Dodatkowo ograniczono zakres wykonywanych przez programy pośredniczące zapytań do: selekcji, dopisywania, modyfikacji danych oraz możliwości wycofywania i zatwierdzania transakcji.

W przyjętym rozwiązaniu nie istnieją również problemy długotrwałych transakcji, które poprzez blokowanie danych utrudniałyby współbieżny dostęp do danych wielu użytkowników, gdyż połączenie do bazy danych (a w trakcie jego trwania wszystkie wykonywane operacje) nawiązywane jest tylko na czas generacji dynamicznej strony WWW, a w momencie przesyłania danych do klienta połączenie jest już zamknięte (i zakończone wszystkie transakcje, a więc odblokowane wszystkie zasoby z nimi związane). W przypadku udostępniania użytkownikom modyfikacji danych na podstawie informacji prezentowanych na poprzednich stronach (typowym przykładem mogą być systemy rezerwacji miejsc) należy tylko dodatkowo programowo uchronić się przed problemem utraconej aktualizacji. Problem ten zagraża systemowi, gdyż dane w czasie między ich prezentacją a próbą aktualizacji mogły się zmienić (nie są dostępne transakcje rozpoczynane na jednej, a kończone na innej stronie). Problem ten nie jest trudny do rozwiązania, gdyż w bazach danych na serwerze SQLBase zaimplementowany jest system unikalnych identyfikatorów wierszy, których zawartość zmienia się wraz z każdą modyfikacją danych w obrębie wiersza. Zapamiętanie wartości takiego identyfikatora w momencie wyświetlania danych i przekazanie tej wartości na stronę modyfikującą dane, która sprawdza istnienie wiersza o podanym identyfikatorze, zabezpiecza przed powstaniem sytuacji błędnej (na przykład podwójnej rezerwacji tego samego miejsca).

LITERATURA

1. Jamsa K., Lalani S., Weakley S.: Programowanie WWW. ZNI Mikom, Warszawa 1997.
2. Castro E.: Po prostu HTML. Helion, Gliwice 1996.
3. SQLBase C Application Programming Interface. Gupta Technologies Inc., 1991.
4. Boring J., Gerber D.: SQLBase Advanced Topics Guide. SQL Designs, Inc. 1995.

5. Dokumentacja elektroniczna: PHP/FI Version 2.0.
6. Dokumentacja elektroniczna: Postgres 95.
7. Dokumentacja elektroniczna: Apache 1.2.
8. Dokumentacja elektroniczna: HTML 3.2.

Recenzent: Dr hab. inż. Stanisław Wolek Prof. Pol. Rzeszowskiej

Wpłynęło do Redakcji 17 października 1997 r.

Abstract

In the global computer network Internet WWW services are one of the most popular ways of publishing informations. The informations presented in this system are typical stored in static flat text files.

In the aim of making possible easier, more flexible selecting informations with complex searching criteria, safe data updates by end-user and optimization of sending data in the net, one can use SQL-database server as the source of information for WWW.

Functional classification of existing solutions in this domain is presented in the paper. Typical multilevel solution contains Fig. 1. The programm package PHP/FI and possibility of its extending by adding function for accessing choosen database server is discussed. The look of example, dynamic WWW pages, written in PHP/FI script language is presented on Fig.2 - Fig.4.

The comparison of using SQL-database as the source of informations for WWW with traditional, static files by example of PHP/FI package extended to access SQLBase server is made.