

Aleksandra WERNER
Politechnika Śląska, Instytut Informatyki

WYDAJNOŚĆ PRZETWARZANIA ZAPYTAŃ W ŚRODOWISKU SFEDEROWANEGO SYSTEMU IBM DB2

Streszczenie. Jedną z podstawowych architektur implementacyjnych systemu rozproszonych baz danych (RBD) jest tzw. system sfederowanych baz danych, który jest w istocie centralnym repozytorium zawierającym dane, do których są dowiązane dane z innych źródeł. W artykule przedstawiono wyniki testów środowiska 3-węzłowego klastra pod kątem wydajności przetwarzania zapytań SQL. Dla wybranych zapytań analizowano ich plany wykonania, w celu identyfikacji czynników wpływających na decyzje optymalizatora zapytań DB2 odnośnie do wyboru strategii przetwarzania danego zapytania.

Słowa kluczowe: sfederowana baza danych, stowarzyszona baza danych, serwer sfederowany, kompensacja, plan wykonania zapytania, optymalizator, integracja danych, opakowanie

PERFORMANCE OF QUERY PROCESSING IN FEDERATED SYSTEM IBM DB2

Summary. One of the basic architecture implementing the Distributed Relational Database Management System is the system of federated database, that is central repository of data originated from different, relational and not relational, sources. The examining platform was 3-node DB2 cluster, where queries efficiency was analyzed. A lot of SQL queries were executed and chosen queries and their execution plans were analyzed, in order to identify query optimizer determinants, affecting query processing strategy.

Keywords: federated database, federated server, compensation, access plan, query optimizer, Information Integrator, wrapper

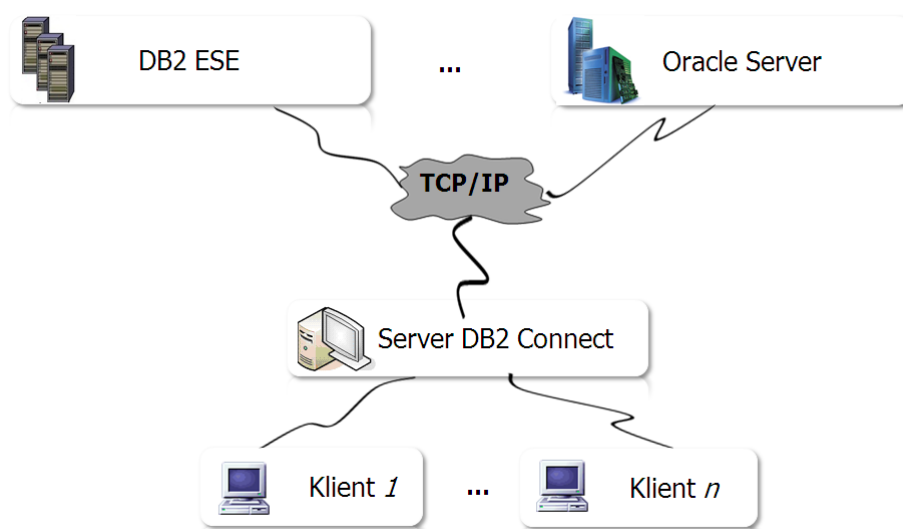
1. Wprowadzenie

W typowych zastosowaniach systemów baz danych wykorzystuje się architekturę scentralizowaną, w której zarówno system zarządzania bazą danych (SZBD), jak i wszystkie dane znajdują się w tym samym węźle sieci informatycznej. Dostęp do takiej bazy danych jest realizowany albo za pomocą aplikacji pracujących w architekturze klient-serwer, albo pracujących w architekturze 3-warstwowej. Istnieje jednak wiele zastosowań, w których scentralizowane bazy danych nie zapewniają wymaganej funkcjonalności i efektywności pracy. Obserwuje się to np. w przedsiębiorstwach czy instytucjach, które są rozproszone pod względem geograficznym, a co za tym idzie, również i ich dane mają charakter rozproszony. Równie często pewne braki czy niedogodności scentralizowanych systemów uwidaczniają się w sytuacji, gdy gromadzone i przetwarzane dane są heterogeniczne – tzn. mają różną strukturę (np. XML, dokumenty tekstowe, arkusze kalkulacyjne) i wykorzystują różne modele danych (np. relacyjne i relacyjno-obiektowe). Podobny brak żądanej spójności obserwuje się nawet wtedy, gdy jedynymi źródłami danych są systemy relacyjne, gdyż bazy danych pochodzące od różnych producentów mogą posiadać różną funkcjonalność, reprezentację danych i dialekt języka SQL. Z tego względu we wszystkich wymienionych przypadkach zachodzi potrzeba zastosowania rozproszonych baz danych, które zintegrują potrzebne informacje.

Jedną z podstawowych architektur implementacyjnych systemu rozproszonych baz danych jest tzw. system sfederowanych (stowarzyszonych) baz danych. Jest to system składający się z co najmniej dwóch niezależnych i autonomicznych systemów baz danych (mogą to być systemy heterogeniczne) oraz odpowiedniego mechanizmu konsolidującego wszystkie ich komponenty, dostarczanego przez serwer sfederowany. Uogólniając, federację można uważać za centralne repozytorium zawierające dane, do których są dowiązane dane z innych źródeł.

Analizowaną architekturę integracyjną, umożliwiającą obustronny dostęp pomiędzy bazami danych Oracle11g i IBM DB2 w wersji Enterprise, przedstawiono na rysunku 1.

Wzajemna komunikacja pomiędzy bazami danych jest tu realizowana za pomocą oprogramowania DB2 Connect Server, które spełnia rolę tzw. bramy. Oprogramowanie to jest w istocie serwerem połączeń, implementującym architekturę DRDA (ang. *Distributed Relational Database Architecture*), który koncentruje połączenia z wielu klientów i aplikacji do wskazanych serwerów, przekazując im instrukcje SQL oraz interfejsy programowania aplikacji API, a także zarządza tymi połączeniami [2]. Opisywana architektura wykorzystuje jako swoje medium transmisyjne protokół TCP/IP.



Rys .1. DB2 Connect Server jako dedykowane oprogramowanie, konsolidujące heterogeniczne bazy danych

Fig. 1. DB2 Connect Server as a dedicated software, consolidating the heterogeneous databases

Przedstawione na rysunku i wykorzystane później w badaniach rozwiązanie można uważać za przykład ściśle skojarzonych federacyjnych baz danych (ang. tightly coupled), które w odróżnieniu do słabo skojarzonych baz danych (ang. loosely coupled), mają serwer zarządzający całością federacji, w którym zdefiniowano schemat federacyjny, będący sumą schematów eksportowych¹ dowiązanych baz danych.

2. Sfederowane środowisko DB2

Środowisko DB2 jest określone przez swoje instancje, zmienne środowiskowe i rejestrowe oraz parametry konfiguracyjne, które są zdefiniowane zarówno na poziomie instancji, jak i bazy danych.

Pojedyncza instancja DB2 jest niezależnym środowiskiem, w którym można tworzyć zarówno bazy danych, jak i ich obiekty oraz uruchamiać aplikacje. Z punktu widzenia architektury, instancja jest warstwą między kodem wykonawczym DB2 a obiektami bazy danych użytkownika, realizującą powiązanie kodu DB2 z tymi obiektami. Wewnętrzna organizacja instancji przypomina drzewo, gdyż są w niej hierarchicznie zdefiniowane obiekty bazy danych. Obiektem najwyższego poziomu jest DB2, który reprezentuje domyślną instancję utworzoną podczas instalacji produktu DB2.

¹ Lokalna baza danych udostępnia aplikacjom działającym na federacji baz danych tylko ściśle określoną część swoich danych (np. wybrane tabele).

Dana instancja może zarządzać jedną lub wieloma bazami danych, z których każda może być osadzona na pojedynczym komputerze albo fizycznie lub logicznie podzielona na kilka serwerów (tzn. spartycjonowana).

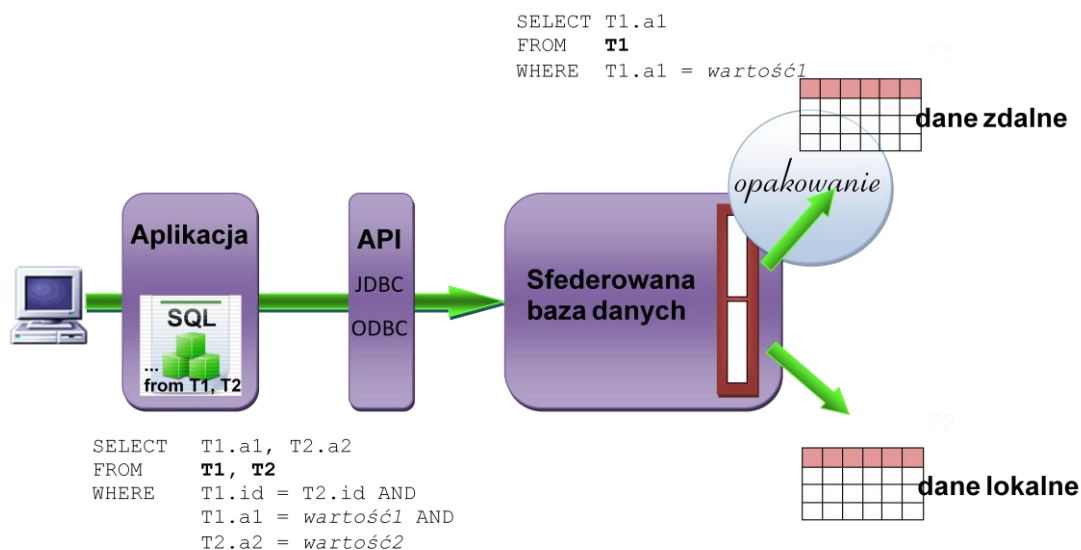
Serwer DB2 może pracować w środowisku sfederowanym i jest wówczas nazywany serwerem sfederowanym. W środowisku tym możliwe jest przetwarzanie danych przechowywanych na innych serwerach i w innych relacyjnych systemach baz danych oraz wysyłanie rozproszonych żądań do wielu źródeł danych przy użyciu pojedynczej instrukcji SQL. Konfiguracja takiego środowiska obejmuje:

- konfigurację instancji bazy danych DB2 zarządzającej systemem sfederowanym, która będzie pełniła rolę sfederowanego serwera,
- dodanie bazy danych, która będzie pełniła rolę sfederowanej bazy danych,
- dodanie jednego lub kilku źródeł danych.

W celu sfederowania źródeł danych będących instancjami baz danych innych niż IBM producentów (np. Oracle, MS SQL Server) lub źródłami nierelacyjnymi (np. dokumenty XML, skoroszyty Excel), konieczne jest dodatkowo zainstalowanie produktu InfoSphere Federation Server, którego zadaniem jest ukrywanie fizycznych własności źródeł danych i przedstawienie ich w sfederowanej bazie w postaci relacyjnych tabel DB2. W przypadku rozbieżności między charakterystyką sfederowanej bazy danych a charakterystyką źródła danych (np. różna strona kodowa), wszelkie przekształcenia dokonywane są zgodnie z właściwościami sfederowanej bazy danych.

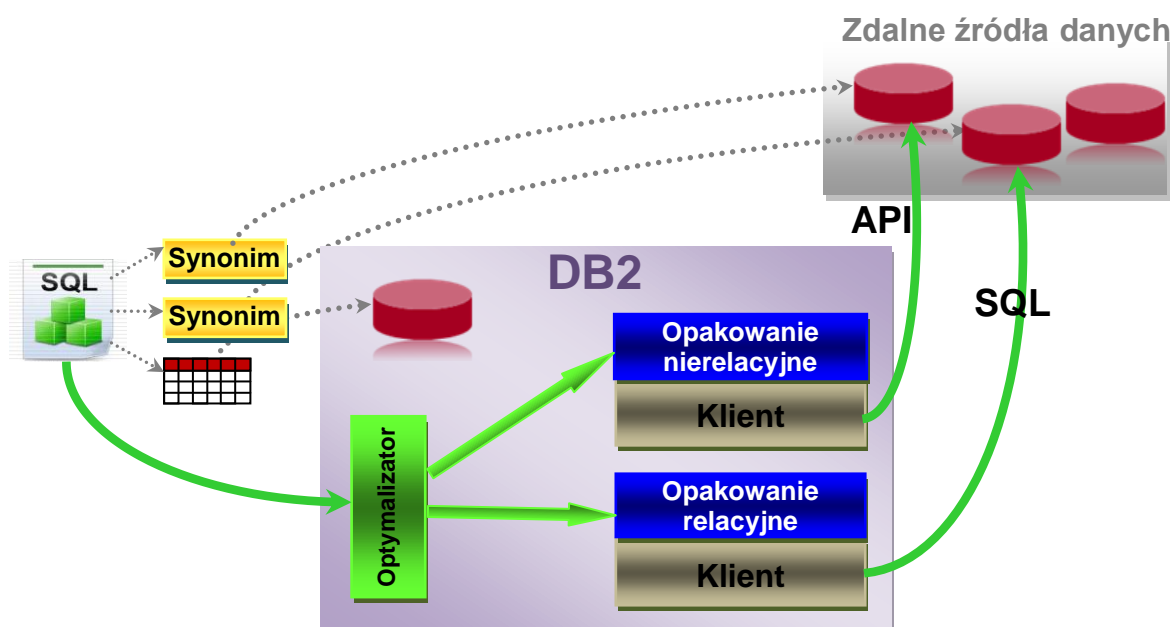
Serwer sfederowany pośredniczy w przekazywaniu żądań kierowanych przez użytkowników i aplikacje klienckie do źródeł danych, przy czym podstawowe algorytmy przetwarzania zapytań w homogenicznych i federacyjnych relacyjnych bazach danych są podobne. Polegają one na dekompozycji zapytania na pewne podzapytania kierowane do lokalnych baz danych, a następnie na skomponowaniu globalnego wyniku z wyników cząstkowych zwróconych przez lokalne bazy danych (rys. 2). W związku z tym, że serwer sfederowany komunikuje się ze źródłami danych za pośrednictwem procedur zapisanych w bibliotece nazywanej modułem opakującym, na rysunku jawnie zaznaczono istnienie mechanizmu opakowania (ang. wrapper).

Z punktu widzenia użytkownika końcowego, zadającego zapytanie do sfederowanej bazy danych, dane z dołączonych lokalnych bądź zdalnych baz są dostępne w taki sposób, jakby były zapisane w jednej dużej bazie danych, z tym że w zapytaniach muszą być używane nazwy zastępcze (synonimy, pseudonimy) tabel sfederowanych – tzw. nicknames.



Rys .2. Dekompozycja zapytania sfederowanego
 Fig. 2. Federated query decomposition

Tak więc aby uzyskać dane np. z 3 źródeł danych, do stowarzyszonej bazy danych wprowadzane jest zapytanie w języku SQL programu DB2, po czym kompilator języka SQL w programie DB2 analizuje na podstawie wpisów w katalogu globalnym i opakowań źródeł danych, jakie operacje muszą być wykonane zdalnie (rys. 3). Optymalizator uwzględnia przy tym również możliwości zdalnego źródła danych.



Rys .3. Idea realizacji zapytań sfederowanych
 Fig. 3. Federated query flow idea

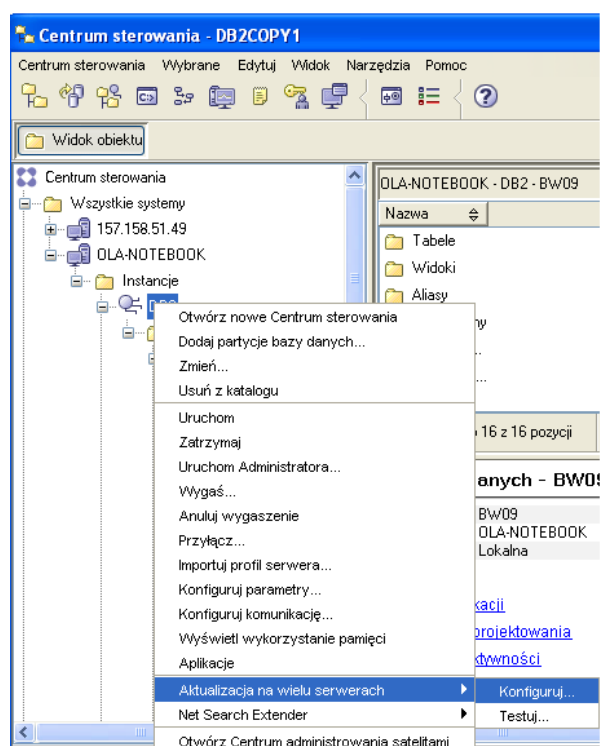
Warto podkreślić, że federacja może przetwarzać lokalne zasoby tylko poprzez interfejs programowania aplikacji (API) specyficzny dla lokalnego systemu. Inne metody (np. bezpośredni dostęp do plików) są niedozwolone.

Widoczne na rysunku strzałki przerywane wskazują na lokalizację źródła danych każdego z obiektów, którego nazwa pojawiła się w analizowanym zapytaniu.

3. Konfiguracja środowiska badawczego

Pierwszym krokiem realizacji prac było skonfigurowanie środowiska badawczego, w skład którego wchodziły 3 komputery osobiste z zainstalowanym systemem Windows XP Professional, połączone ze sobą siecią Ethernet, pracującą pod kontrolą protokołu TCP/IP.

Na ww. elementach systemu uruchomiono instancje, a następnie za pośrednictwem Centrum Sterowania utworzono – zgodnie z wytycznymi w [3] – dla instancji DB2 monitor przetwarzania transakcyjnego TM (rys. 4).



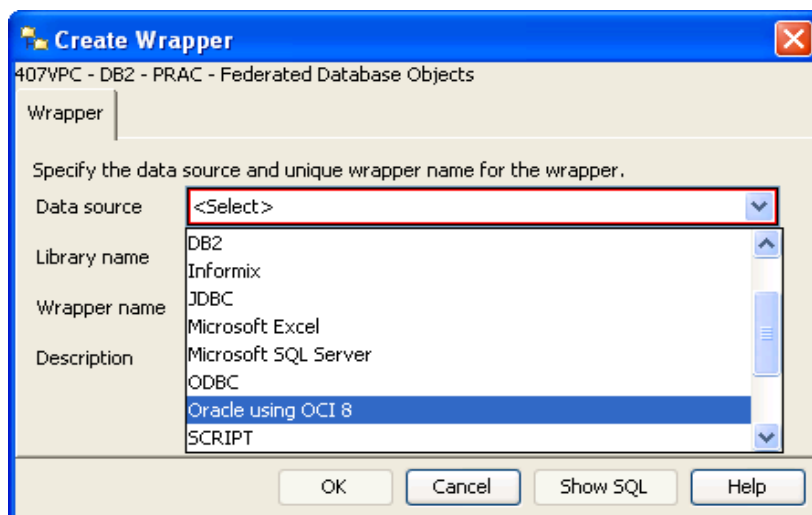
Rys .4. Uruchamianie kreatora konfiguracji systemu rozproszonego
Fig. 4. Activation of distributed system configuration creator

W następnej kolejności powołano do życia obiekty typu wrapper (opakowanie) i zarejestrowano je [3 i 4] (rys. 5). Na każdy typ źródła danych, do którego wymagany był dostęp, utworzono po jednym opakowaniu.

Przykładowo opakowanie DRDA:

```
CREATE WRAPPER DRDA LIBRARY 'db2drda.dll';  
CREATE SERVER TOOLSDB TYPE DB2/UDB VERSION '9.1'  
WRAPPER DRDA AUTHID "admin" PASSWORD "*****" OPTIONS( ADD DBNAME 'TOOLSDB');
```

umożliwiło dostęp do wszystkich obiektów źródeł danych z rodziny DB2.



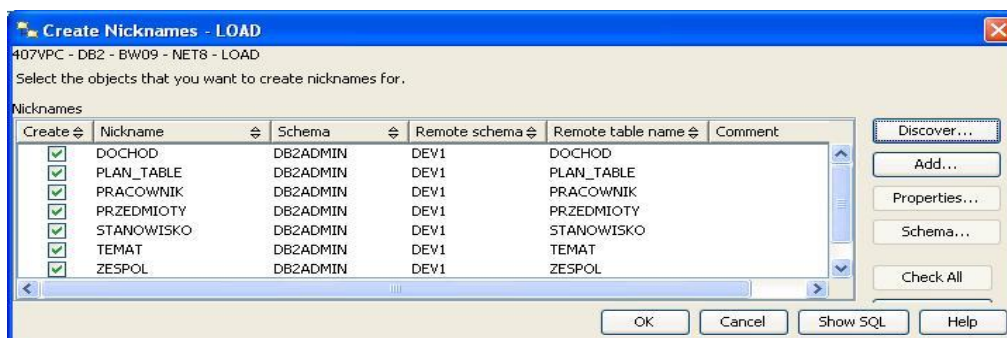
Rys .5. Tworzenie opakowania dla źródła danych Oracle
Fig. 5. Oracle data source wrapper creation

Wybrana instancja pracowała jako serwer sfederowany – w tym celu ustawiono dla niej zmienną `FEDERATED`:

```
UPDATE DBM CFG USING FEDERATED YES;
```

i zrestartowano poleceniami `DB2STOP FORCE` oraz `DB2START`.

Ostatnim etapem konfiguracji sfederowanej bazy danych było utworzenie synonimów do tabel zdalnych (rys. 6). Następnie wykonywano już polecenia SQL (zapytania) na danych znajdujących się w zdalnych/lokalnych bazach danych.



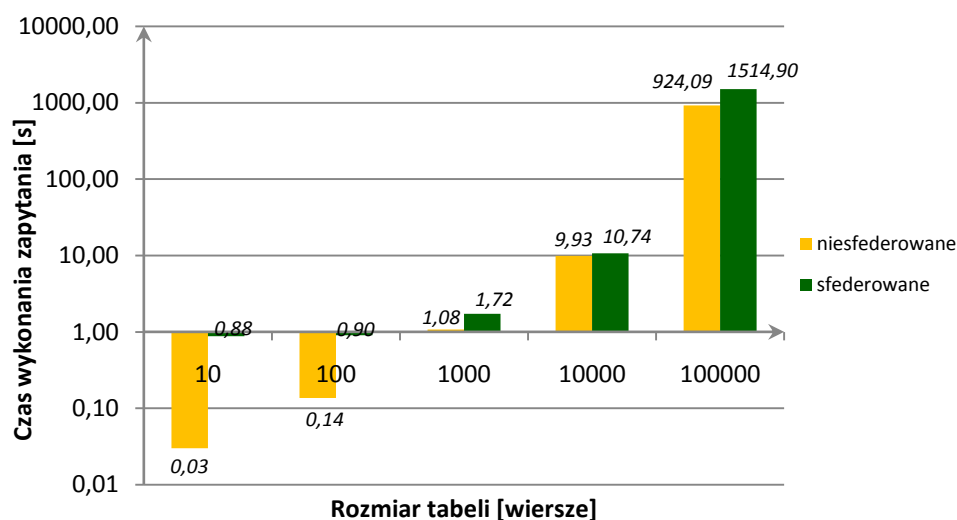
Rys .6. Tworzenie synonimów do wybranych tabel systemu Oracle
Fig. 6. Oracle tables nicknames creation

3.1. Przykładowe testy wydajnościowe

W celu przetestowania wydajności wyszukiwania w bazach danych przeprowadzono kilka eksperymentów. Pierwszy polegał na porównaniu szybkości wykonania zapytań łączących tabele należące do lokalnej, niesfederowanej bazy danych i tabele należące do zdalnej, sfederowanej bazy danych. Tabele miały zmienny rozmiar – od 10 do 100 tysięcy wierszy. Czasy wykonania zapytań mierzono poleceniem:

```
db2batch -d <nazwa_bazy_danych> -f <nazwa_pliku>.sql
```

Zaobserwowano, że wraz ze wzrostem liczby wierszy przetwarzanych i zwracanych przez zapytanie, różnica pomiędzy czasem wykonania zadania na tabelach niesfederowanych (lokalnych) i sfederowanych (zdalnych) malała (rys. 7²). Decydującą poprawę dało się zauważyć już dla tabel o liczbie wierszy sięgającej kilku tysięcy.



Rys. 7. Przyrost czasu wykonania zapytania dla 2 tabel o zmiennej liczbie wierszy
Fig. 7. 2-tables query execution time increase for varying number of processed rows

Stwierdzono przy tym, że wraz ze wzrostem liczby tabel łączonych w zapytaniu, zauważona poprawa była bardziej znacząca (rys. 8).

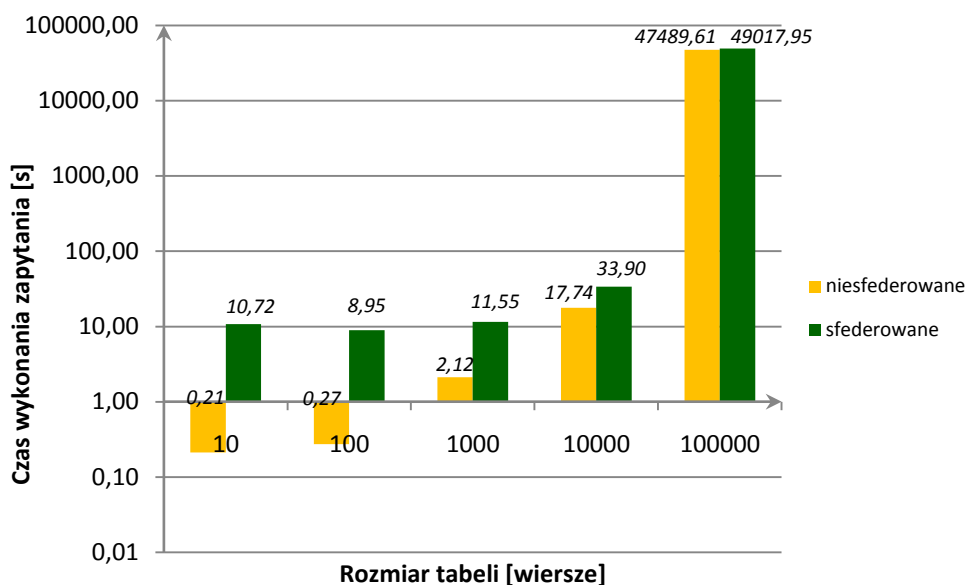
I tak np. dla 2 tabel łączonych w zapytaniu o licznosci 10 wierszy, stosunek czasu wykonania zadania na bazie zdalnej do czasu wykonania zadania na bazie lokalnej wynosił nieco ponad 29, podczas gdy stosunek ten dla analogicznego zapytania operującego na 4 tabelach wynosił już ponad 50³. Z kolei dla liczby wierszy równej 10 tysięcy, przyrost czasu wynosił odpowiednio:

- dla tabel lokalnych – około 1,1,

² Na wszystkich wykresach, na osi pionowej (czas wykonania zadania) zastosowano skalę logarytmiczną.

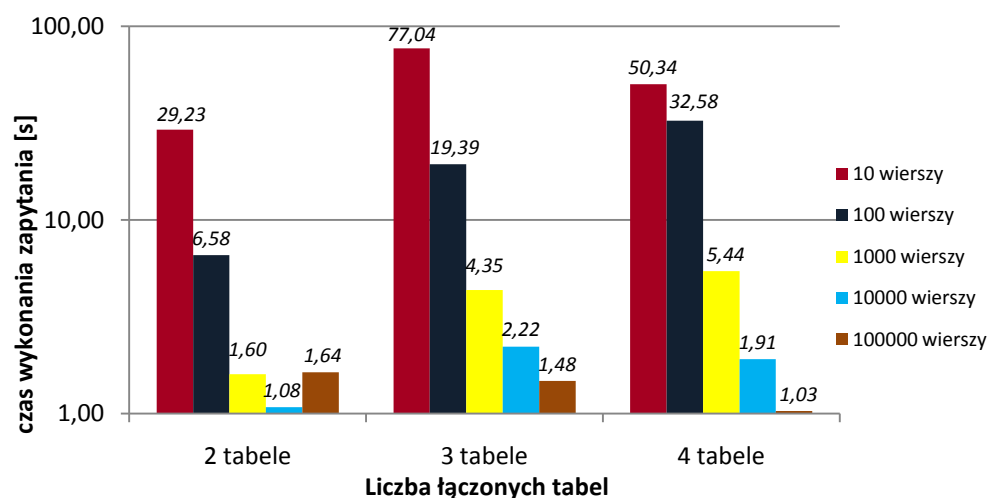
³ Tzn. czas wykonania zapytania sfederowanego był ponad 50 razy wolniejszy od tego, który był wykonywany na bazie lokalnej.

- dla tabel zdalnych – niecałe 2.



Rys .8. Przyrost czasu wykonania zapytania dla 4 tabel o zmiennej liczbie wierszy
 Fig. 8. 4-tables query execution time increase for varying number of processed rows

W celu lepszego zobrazowania działania optymalizatora, na rysunku 9 zestawiono wyniki uzyskane podczas realizacji zapytań kierowanych do sfederowanej bazy danych.



Rys .9. Przyrost czasu wykonania zapytania operującego na sfederowanych tabelach o zmiennej liczbie wierszy
 Fig. 9. Federated query execution time increase for varying number of tables rows

Aby móc dogłębnie przeanalizować sposób przetwarzania zapytań kierowanych do sfederowanej bazy danych, w następnej kolejności śledzono plany wykonania wybranych zapytań, wydając w tym celu polecenie:

```
db2expln -d <nazwa_bazy_danych> -t -z @ -f <nazwa_pliku>.sql.
```

Dodatkowo, mając na uwadze eliminację ewentualnego korzystnego działania innych czynników na czas realizacji zapytania, w przetwarzanych tabelach nie zakładano żadnych indeksów, które w przypadku zapytań należących do niektórych klas (tzw. zapytania o klucz), mogły narzucać błędne wnioski o efektywności działania optymalizatora sfederowanej bazy danych.

Wygenerowane plany wskazały m.in., że dany predykat jest wartościowany zdalnie wtedy, kiedy ma charakter selektywny i przyczynia się tym samym do zmniejszenia ruchu w sieci. W połączeniu z ustawieniem opcji serwera `COMM_RATE`⁴ na odpowiednio małą wartość (np. 1 lub 2), optymalizator istotnie tworzy zapytanie, które generuje minimalny ruch w sieci.

4. Podsumowanie

W ramach prezentowanej pracy przeanalizowano wpływ architektury sfederowanych baz danych na wydajność przetwarzanych zapytań. W tym celu skonfigurowano dwa niezależne systemy – system lokalnej bazy danych DB2 oraz system rozproszonej sfederowanej bazy danych, w których następnie utworzono analogiczne tabele o identycznej liczbie wierszy. Przeprowadzone testy pozwoliły stwierdzić, że federacja nie ma wpływu na algorytmy przetwarzania transakcji stosowane przez lokalne bazy danych. Efektem testów było również potwierdzenie przypuszczenia, że optymalizator może zdecydować o zaniechaniu wykonania bezpośrednio w zdalnym źródle, jeśli okaże się to nieekonomiczne. Tak więc ostateczny plan dostępu wybrany przez optymalizator może składać się z operacji w zdalnym źródle danych, ale czasem podejmuje on decyzję o wykonaniu zadań bezpośrednio w serwerze stowarzyszonym, sprowadzając (przepisując) w tym celu dane ze źródła do węzła lokalnego. Jest to tzw. kompensacja.

Warto podkreślić, że optymalizator zapytań, opracowując alternatywne strategie przetwarzania zapytań, uwzględnia jeszcze inne czynniki – np. takie, jak:

- objętość danych, które mają być przetworzone,
- szybkość przetwarzania danych przez źródło danych,
- objętość danych, które fragment zapytania ma zwrócić,
- przepustowość łącza komunikacyjnego [6].

Stwierdzono ponadto, że istnieje możliwość ręcznego skonfigurowania optymalizatora. Na przykład w celu wymuszenia przekazywania przez optymalizator możliwie największej

⁴ Jest to parametr będący liczbą całkowitą, wyrażony w [MB/s], który określa szybkość komunikacji między serwerem stowarzyszonym a serwerem źródła danych.

liczby operacji do źródła danych, parametr `DB2_MAXIMAL_PUSHDWN` powinien przyjąć wartość x . Wówczas podstawowym kryterium optymalizatora będzie nie koszt, ale ruch w sieci.

BIBLIOGRAFIA

1. Opis standardu DRDA: DRDA V4 Vol. 1, 2, 3 – Distributed Relational Database Architecture. Online Bookstore. Adres strony internetowej (stan na rok 2009): <http://www.opengroup.org/dbiop/>.
2. Podręcznik użytkownika DB2 Connect: ConnectIBM® DB2 Connect™ User's Guide 9.5. Adres strony internetowej (stan na rok 2009): ftp://ftp.software.ibm.com/ps/products/db2/info/vr95/pdf/pl_PL/db2c0p952.pdf.
3. Centrum Informacyjne DB2. Adres strony internetowej (stan na rok 2009): <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp>.
4. Sprenglewska J., Sprenglewski J.: Rozproszone systemy transakcyjne na przykładzie bazy DB2 UDB v8.1. GazetaIT, 2003. Adres strony internetowej (stan na rok 2009): <http://gazeta-it.pl/200305225926/Rozproszone-Systemy-Transakcyjne-na-Przykladzie-Bazy-DB2-UDB-v8.1.html>.
5. R. F. Chong, C. Liu, S. F. Qi, D. R. Snow: Zrozumieć DB2: Nauka na przykładach, MIKOM, 2006 r.
6. Materiały IBM: Podręcznik systemów stowarzyszonych, SC85-0104-01, 2006 r. Adres strony internetowej (stan na rok 2009): ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/pl_PL/iifyfpp81.pdf

Recenzent: Dr inż. Dariusz Mrozek

Wpłynęło do Redakcji 29 listopada 2009

Abstract

Federated database is a heterogeneous database system that provides a single database image of multiple databases. Because of the fact that data can originate from different resources (e.g. relational or unrelational), it is sometimes needed to install federation server, that integrates them into one federated system. The database clients access remote data through user defined nicknames, but this architecture is completely transparent.

Described researches were focused on analyzing federated queries efficiency. To pursue a goal, a lot of SQL queries were executed and chosen queries and their execution plans were analyzed. This allowed to identify query optimizer determinants, affecting query processing strategy. Besides, several existing configuration parameters were used to configure federated server.

The examining environment was composed of 3 personal computers with Windows XP Professional System. DB2 Connect Server was installed on chosen node so federated server run exactly there. In order to do intended researches, Oracle and DB2 databases were federated into the one system.

Adres

Aleksandra WERNER: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, aleksandra.werner@polsl.pl .