

Alicja Kilińska – Wiśniewska
Politechnika Koszalińska Wydział Informatyki i Elektroniki

Pomiar czasu przetwarzania rozproszonych zapytań w ewoluujących silnikach serwerów baz danych SQL Microsoft Server.

Słowa kluczowe: zapytanie rozproszone, czas przetwarzania zapytania.

1. Wstęp

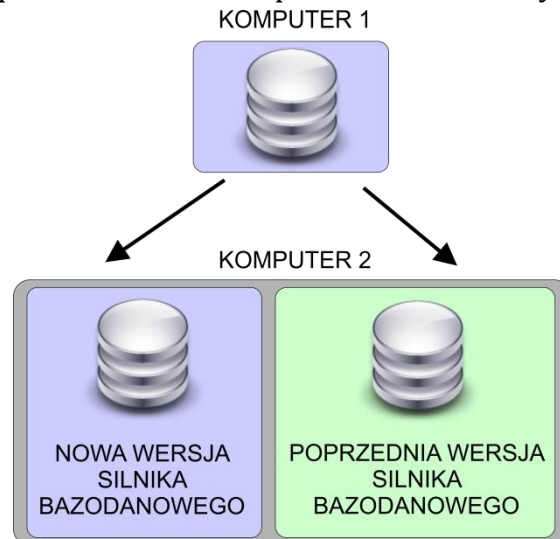
Organizacja, przechowywanie oraz szybki dostęp do danych umieszczonych w bazach danych jest obecnie jedną z podstawowych idei rozwoju współczesnej informatyki. Wraz ze zwiększonymi wymaganiami systemowymi wzrasta zapotrzebowanie na przechowywanie danych które te systemy generują lub wykorzystują. Dane wzrastają w tempie wykładniczym, co doskonale widać na przykładzie nieustannie rosnących pojemności dysków twardych, które w latach 90 – tych osiągały pojemności rzędu 4,2 Gb, natomiast obecnie dochodzą one do rozmiarów 1 Tb.

Jednocześnie wymaga się coraz szybszego bardziej elastycznego oraz nieprzerwanego dostępu do danych przechowywanych w bazach danych. Dynamiczny rozwój infrastruktury sieciowej, coraz większe możliwości szybkiego przesyłania informacji, współdzielenie i udostępnianie danych spowodowało, że coraz większą rolę zaczęły odgrywać rozproszone bazy danych. Doskonałym przykładem zastosowania rozproszonych baz danych na skalę światową okazała się akcja SETI, która wykorzystywała zasoby komputerów użytkowników rozsianych na całym świecie za pośrednictwem sieci Internet. Była ogromnym sukcesem, aczkolwiek pokazała jak istotny jest czas dostępu do danych rozsianych po całym świecie. Obecnie pracuje się nad przyspieszeniem oraz zabezpieczeniem dostępu do danych przechowywanych w rozproszonych bazach danych.

2. Motywacja

Powszechność rozproszonych baz danych sprawia iż ilość danych przechowywanych w bazach tego typu rośnie bardzo szybko. Wymusiło to na producentach oprogramowania, a dokładniej silników bazodanowych ciągłe modyfikacje istniejących wersji serwerów bazodanowych oraz zbyt częste publikowanie nowych wersji oprogramowania. Korporacje producenckie upatrują w tym łatwy i szybki zysk, gdyż powszechnie przyjętą zasadą jest zakupywanie nowych wersji oprogramowania które pojawia się na rynku. Jednakże istnieje możliwość iż rozproszona baza danych znajdująca się na kilku serwerach różniących się pomiędzy sobą wersjami silnika nie będzie funkcjonowała spójnie co doprowadzi do spowolnienia działania bazy danych oraz obniży wydajność samej rozproszonej bazy danych.

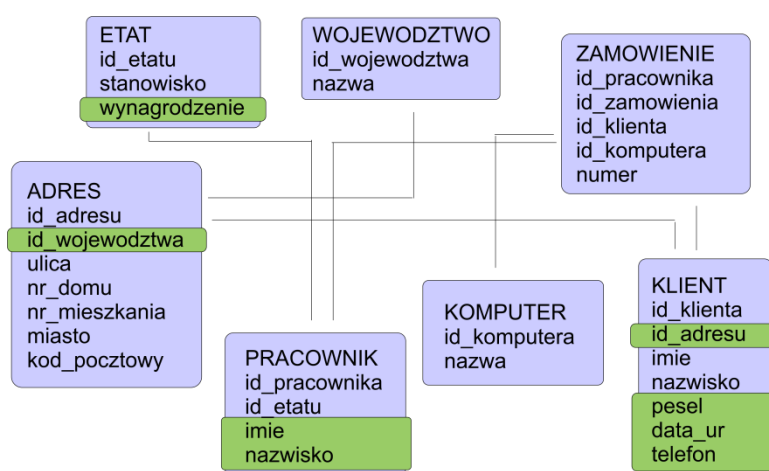
Rozpatrzmy zatem następujący przykład rozproszonej bazy danych która została zaimplementowana na dwóch silnikach bazodanowych jednego producenta oprogramowania w wersji starszej oraz nowszej. Wspomniane dwie wersje oprogramowania zostały zainstalowane na komputerze położonych w sieci, miało to na celu wykluczenie czynnika różnorodności podzespołów zainstalowanych w samym komputerze, które mogłyby wpłynąć na proces badania. Baza została rozproszona pomiędzy dwoma komputerami co zostało przedstawione na rysunku 1.



Rys. 1. Schemat rozproszenia badanej bazy danych.

Umieszczony na rysunku 1 schemat przedstawia rozproszenie badanej bazy danych pomiędzy dwoma komputerami umieszczonymi w sieci, które posłużyły do przeprowadzenia badania. Warto podkreślić, że komputery znajdowały się w jednej sieci, która została stworzona specjalnie na potrzeby przeprowadzenia badania, aby wykluczyć czynniki zewnętrzne które mogły być mieć wpływ na wynik przeprowadzonego badania.

Baza danych rozstała rozproszona zgodnie ze schematem zamieszczonym na rysunku 2.



Rys. 2. Schemat badanej, rozproszonej bazy danych.

Umieszczony na rysunku 2 schemat pokazuje reprezentację głównych elementów opisywanej bazy danych: *etat*– *województwo*– *zamówienie* – *adres* – *pracownik* – *komputer* - *klient*. Ten schemat został przedstawiony jako model fizyczny. Baza została podzielona pomiędzy dwoma silnikami bazodanowymi znajdującymi się na

dwóch komputerach. Część bazy danych oznaczona na kolor fioletowy została umieszczona na komputerze numer 1, natomiast część oznaczona na kolor zielony została umieszczona na komputerze numer 2.

Aby uzyskać pełne dane klienta zamawiającego pewien produkt należy odwołać się do tabeli *klient* która została rozproszona.

3. Mechanizm przetwarzania zapytań w bazach danych

Istotnym elementem na który należy zwrócić uwagę jest schemat przetwarzania zapytania przez wykorzystane silniki bazodanowe. W tym wypadku zostanie do tego wykorzystana instrukcja SELECT języka SQL. Zapytania pod względem przetwarzania można podzielić na zapytania ad hoc oraz składowane, które dodatkowo można podzielić na etapy w których są wykonywane. Zapytanie ad hoc składa się z etapów: poprawność syntaktyczna, poprawność semantyczna, optymalizacja zapytania, kompilacja i wykonanie zapytania. Natomiast zapytanie składowe określane jest jako kolejne wywołanie tego samego zapytania. Tego typu zapytania wykonywane są jedynie w dwóch etapach składających się z pobrania skompilowanej instrukcji oraz jej wykonania.

Instrukcja SELECT została wykorzystana, aby przy jej pomocy dokonać pomiarów czasu jej wykonania na rozproszonej bazie danych. Sama instrukcja służy do pobrania pożądaných danych z bazy. Dodatkowo została wykorzystana również instrukcja FROM, która definiuje z której tabeli lub innego elementu takiego jak widok mają zostać pobrane dane.

4. Pomiar czasu statystycznego

Celem dokonania pomiaru czasu statystycznego było udowodnienie założonej tezy iż starszy silnik bazodanowy znacznie wolniej przetwarza zapytanie rozproszone niż nowszy, co miało ogólny wpływ na szybkość otrzymywania wymaganych wyników. Do tego celu został wykorzystany wyżej opisany model rozproszonej bazy danych rozmieszczonej pomiędzy komputerami w sieci. Mierzony był czas statystycznych podawany przez serwer podczas przetwarzania zapytania. Pomiar statystyczny został w zakresie analizy, kompilacji i czas realizacji transakcji, które występują po stronie serwera, otrzymane czasynie uwzględniając czas potrzebnego na przesyłanie danych do klienta, co wyklucza błąd pomiaru wynikający z obciążenia sieci komputerowej.

Podczas badania uwzględnione zostały dwie wartości czasowe. Pierwszym wynikiem dotyczył czasu parsowania i czasu kompilacji zapytania, natomiast drugi wynik był czasem analizy i opracowanie odnoszącym się do czasu opracowania całego zapytania. Obie wartości określają jak długo były wykonywane czynności:

- Przetwarzanie polecenia SQL, sprawdzanie błędów składni, łamiąc się do polecenia części produkcji wewnętrznej drzewo wykonania.
- Kompilacja plan wykonania w pamięci cache oddrzewo realizacji, które właśnie zostały wyprodukowane.

Należy jednak nadmienić, że obejmuje to czas, który jest wymagany do przenoszenia danych z tabel powiązanych w pamięci podręcznej, dlatego jeśli wykonane będzie to samo zapytanie dwa razy, a dane z tabeli będą nadal znajdowały się w pamięci podręcznej, to czas parsowania i czasu kompilacji będą znacznie niższe podczas drugiego wykonania zapytania.

*SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.*

Zajętość czasu procesora 0ms i 0ms upływu czasu na zakończenie przetwarzania i kompilacji. Upływający czas obejmuje (CPU TIME, IO Time itp.).

*SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.*

To całkowita ilość czasu ile zajęło wykonanie zapytania. Trwało to 0 ms czasu procesora i 0 ms upływającego czasu, co oznacza 0 ms na przetwarzanie IO.

5. Analiza otrzymanych wyników

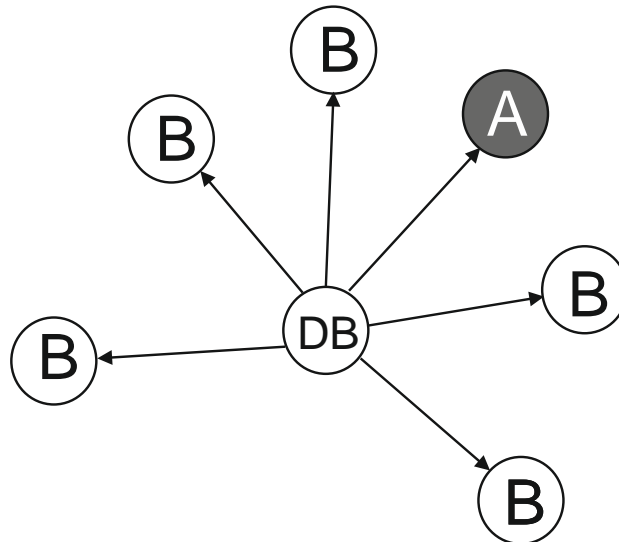
Na podstawie przeprowadzonych badań można jasno poprzeć tezę iż silniki baz danych różniące się tylko jedną wersją wstecz działają o około 3,75 ms wolniej przy uwzględnieniu wszystkich czasów pomiaru statystycznego.

Średni czas zajętości procesora podczas parsowania zapytania wynosił średnio 0 ms dla baz z nowym silnikiem i 3,85 ms dla baz ze starszym silnikiem. Natomiast czas średni czas wykonania IO rozproszonego zapytania wynosił 4,12 ms dla baz z nowym silnikiem i 9,5 ms dla wersji starszej, co za tym idzie można łatwo wysnuć wniosek, że kompilacja przebiega o 5,37 ms szybciej przy użyciu nowego silnika bazodanowego.

Całkowite wykonanie zapytania natomiast zajęło procesorowi średnio 1,87 ms przy nowym silniku, przy starym trwało to średnio 3,87 ms, czyli aż o 2 ms dłużej. Jednakże najistotniejszy był czas wykonania i kompilacji całego zapytania rozproszonego, gdyż tutaj również do porównania został zmierzony czas na lokalnym serwerze. Z pomiarów wynika iż tylko dla kompilacji całego zapytania czas ten jest większy od 0 ms. Pomiar wykazały również opóźnienie względem starszej wersji silnika, gdyż otrzymane wartości wynosiły odpowiednio: dla nowego silnika 23,25 ms, dla wersji poprzedniej 27,12 ms, a dla zapytania wykonanego stacjonarnie 5,12 ms. Różnica pomiędzy wersjami silnika wyniosła aż 3,87 ms, wartość ta oczywiście wraz z rozrostem bazy danych najprawdopodobniej ulegnie zwiększeniu.

Rysunek 3 przedstawia przykładowy graf rozproszenia bazy danych. Wierzchołek DB traktowany jest głównym serwerem zarządzającym całą bazą, natomiast poszczególne węzły to części rozproszone bazy danych na innych serwerach. Jako zaciemniony wierzchołek na grafie został przedstawiony serwer ze starszym silnikiem bazodanowym oznaczona literą A, natomiast pozostałe silniki są w wersji nowszej oznaczone literą B.

W tym przykładzie uwzględniony zostanie całkowity czas kompilacji zapytania rozproszonego. Zgodnie z przeprowadzonymi badaniami, jeżeli podczas przetwarzania zapytania, które wymaga wykorzystania danych znajdujących się na wszystkich bazach danych, średni czas kompilacji zapytania na wierzchołku A wyniesie 27,12 ms, natomiast na pozostałych wierzchołkach średni czas będzie wynosił 23,25 ms. Złożenie zapytania na wierzchołku DB jest maksymalnym czasem wybranym ze wszystkich czasów kompilacji na pozostałych wierzchołkach.



Rys. 3. Graf przedstawiający rozproszenie przykładowej bazy danych. Można to zapisać według wzoru:

$$DB = \max(A, B)$$

gdzie A, B – czasy przetwarzania zapytania na poszczególnych wierzchołkach

Zatem otrzymana zostanie wartość najwyższa w tym wypadku 27,12 ms. Przy wykonaniu n zapytań strata czasu wynikająca z opóźnienia generowanego przez wierzchołek A będzie systematycznie rosła i może zostać opisana zależnością:

$$OP = \sum_n n \cdot (\max(A, B) - \min(A, B))$$

Gdzie:

OP - średni czas opóźnień,

n - ilość wykonanych zapytań,

6. Podsumowanie

W pracy przedstawiono wyniki i interpretację badań przeprowadzonych na rozproszonych bazach danych uzależnionych od wersji silnika bazodanowego. Celem przeprowadzonych badań miało być potwierdzenie lub zaprzeczenie teorii iż wszystkie używane silniki baz danych w jednej rozproszonej bazie danych muszą być tej samej generacji, jak również potwierdzenie, że ewolucja silników bazodanowych wpływa na czas przetwarzania zapytań w rozproszonych bazach danych. Otrzymane wyniki jednoznacznie potwierdziły postawione tezy. Należy zwrócić również uwagę iż rosnąca popularność baz danych wymusza konieczność stosowania ujednoliconych silników bazodanowych aby efekty pracy z tymi typami baz danych przynosiły oczekiwane korzyści.

Bibliografia:

„Transact-SQL. Czarna księga” - Marcin Szeliga,

„Principles of Distributed Database Systems – M.TamerOzsu, Patrick Valduriez

Streszczenie

Opracowanie zawiera przegląd metod przetwarzania zapytań w rozproszonych relacyjnych systemach zarządzania bazami danych. Omówione zostały podstawowe

założenia, pojęcia i metody. Założeniem jest ustosunkowanie się do pomiaru czasu przetwarzania zapytań w rozproszonych bazach danych. Opracowanie omawia także samą koncepcję przetwarzania zapytań w rozproszonych bazach danych opartych na silniku serwerów bazodanowych.

Słowa kluczowe: zapytanie rozproszone, czas przetwarzania zapytania.

Measurement of time in query processing engines evolving database servers.

Summary

The study contains an overview of methods for query processing in distributed relational database management systems. Discusses the basic assumptions, concepts and methods. The idea is to comment on the timing of processing queries in distributed databases on the engine. The on database servers.

Keywords: Distributed query, query processing time.