

Juliusz JEZISKI*
Tadeusz MORZY
Mikolaj MORZY

OPTYMALIZACJA ZAPYTAN W SRODOWISKU SIECI ROZLEGLEJ

Algorytmy optymalizacji zapytan w srodowisku sieci rozleglej musza uwzgledniac nowe wymagania dotyczace: dynamicznie zmieniajacych sie opoznien w srodowisku komunikacyjnym oraz autonomii wezlow utrudniajacej zbieranie statystyk i szacowanie kosztu realizacji zapytan. Wymaga sie zastosowania nowych, adaptacyjnych technik optymalizacji umozliwiajacych aktywna reorganizacje sposobu realizacji zapytania zgodnie z fluktuacja charakterystyki danych oraz opoznien srodowiska komunikacyjnego. Artykul przedstawia stan badan nad optymalizacja zapytan w srodowisku rozleglym, w szczegolnosci zostana opisane rodziny algorytmow *Query Scrambling*.

1. WSTEP

Obserwowany od kilku lat rozwoj uslug internetowych umozliwia dostep do danych gromadzonych i przetwarzanych na calym swiecie. Uslugi te sa wykorzystywane w bankowosci, w handlu, w turystyce, w nauce, na gieldzie i wielu innych zastosowaniach.

W chwili obecnej systemy baz danych w malym stopniu wspieraja wyszukiwanie informacji zgromadzonych w Internecie. Silnym ograniczeniem jest brak algorytmow optymalizacji zapytan uwzgledniajacych dynamiczna charakterystyce opoznien w dostepie do zasobow. Zmienne opoznienie jest spowodowane awariami oraz zmienna aktywnoscia uzytkownikow w ciagu doby na calym swiecie. Nowe algorytmy optymalizacji zapytan w takim srodowisku powinny pozwalac na elastyczna reorganizacje planu wykonania zapytan w zaleznosci od aktualnego obciazenia zasobow Internetu.

Celem niniejszego artykulu jest przedstawienie stanu badan nad optymalizacja zapytan w dynamicznym srodowisku Internetu. W szczegolnosci zostanie zaprezentowany zbior algorytmow pod nazwa *Query Scrambling* [1] [2] [3]. W ramach tej prezentacji zostana opisane podstawy teoretyczne, szczegolowe rozwiazania techniczne oraz wyniki eksperymentow symulacyjnych. W podsumowaniu zostana przedstawione dalsze kierunki badan w tej dziedzinie.

2. ALGORYTMY RODZINY *Query Scrambling*

Punktem wyjscia do dzialania algorytmow *Query Scrambling* jest plan zapytania opracowany przez klasyczny algorytm optymalizacji bez uwzglednienia fluktuacji srodowiska

* Instytut Informatyki, Politechnika Poznanska (ul. Piotrowo 3a, 60-965 Poznan,
{Juliusz.Jeziski,Mikolaj.Morzy}@cs.put.poznan.pl, Tadeusz.Morzy@put.poznan.pl)

komunikacyjnego. W celu opracowania wyjściowego planu, w zależności od dostępności informacji o rozkładzie statystycznym danych, mogą być wykorzystywane algorytmy kosztowe oparte na metaheurystykach (np. *Simulated Annealing*, *Tabu Search*) lub reguły oparte o ranking operacji realizujących zapytanie.

Przygotowany w ten sposób plan jest wykonywany przez moduł wyposażony w detektor opóźnień komunikacyjnych. W przypadku wykrycia opóźnień o zadanych parametrach sterowanie przekazywane jest z powrotem do optymalizatora, który może podjąć decyzję zmieniającą plan zapytania w taki sposób aby zminimalizować straty w koszcie realizacji zapytania. Działania optymalizatora są wykonywane na dwóch fazach:

1. **Przeszeregowanie operatorów** polega na zmianie kolejności realizowania poszczególnych kroków planu zapytania bez zmiany kształtu oryginalnego planu,
2. **Synteza operatorów** powoduje zmianę planu oryginalnego planu przez dodanie, usunięcie lub zmianę elementarnych kroków planu zapytania.

2.1. PRZESZEREGOWANIE OPERATORÓW

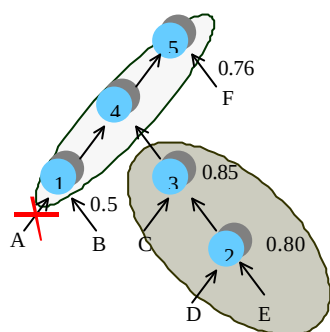
Na etapie przeszeregowania operatorów znajdują się części planu, które nie mogą być wykonane w wyniku powstania opóźnienia, takie fragmenty planu nazywa się *zablokowanymi*. Ponadto optymalizator wyodrębnia fragmenty planu, nazywane *realizowalnymi*, które nie są zablokowane i można je wykonać w trakcie oczekiwania na dane z zablokowanego węzła.

Wykonanie fragmentu planu poza kolejnością wynikająca z oryginalnego planu obciążone jest dodatkowym kosztem związanym z koniecznością przechowywania (materializacji) wyników pośrednich. Wybór odpowiedniego realizowalnego fragmentu planu do wykonania jest podejmowany z uwzględnieniem kosztu materializacji. Wybór ten jest dokonywany na podstawie współczynnika efektywności obliczanego dla każdego realizowalnego fragmentu planu na podstawie zależności: $\frac{P - MR}{P + MW}$, gdzie P jest to koszt wykonania danego

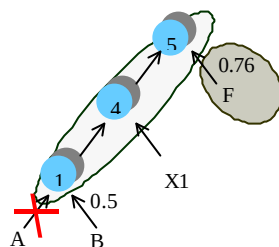
fragmentu planu, MW jest to koszt materializacji danych pośrednich, MR jest to koszt odczytu zmaterializowanych danych. Mianownik tego wyrażenia określa koszt wykonania fragmentu planu wraz z kosztem materializacji wyniku, licznik natomiast wyraża zysk związany z przyszłym wykorzystaniem wyniku wykonania fragmentu planu. Współczynnik efektywności określa względny zysk wykonania wybranego fragmentu planu poza oryginalną kolejnością w stosunku do kosztu wykonania tej operacji. Do realizacji są wybierane te fragmenty, które posiadają największe współczynniki efektywności. W celu uniknięcia nadmiernych kosztów odrzucane są te fragmenty, których współczynnik efektywności spada poniżej progu weta wstępnie ustalonego ona 0,75.

Dla ilustracji fazy przeszeregowania operatorów przeanalizowano przykładowe zapytanie (patrz rys. 1) odwołujące się do sześciu źródeł danych umieszczonych na różnych węzłach rozległego środowiska komunikacyjnego. Węzły oznaczono literami od A do F. Plan zapytania wygenerowanego przez klasyczny optymalizator przedstawiono w postaci drzewa, gdzie liście ilustrują źródła danych, krawędzie przepływ danych w środowisku komunikacyjnym, a pozostałe węzły operacje połączenia. Oznaczenia liczbowe w okręgach określają kolejność wykonywania operacji.

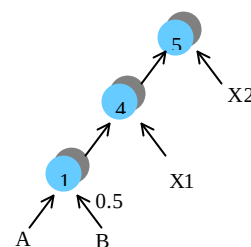
W wyniku opóźnienia w dostępie do danych z węzła A optymalizator podejmuje decyzje o przeszerzegowaniu operatorów. W celu znalezienia optymalnego rozwiązania znajdowany jest zablokowany fragment planu (na rysunku zaznaczony jasnym cieniowaniem) oraz wszystkie realizowalne fragmenty planu. Następnie dla wszystkich realizowalnych fragmentów określany jest współczynnik efektywności i do realizacji jest wybierany fragment o najwyższym współczynniku (na rysunku zaznaczony ciemnym cieniowaniem). Jeżeli węzeł A (patrz rys. 2) nie został w tym czasie odblokowany, to optymalizator wyszukuje do realizacji kolejny fragment planu. Fragment planu związany z pobraniem danych z węzła B nie zostanie wzięty pod uwagę ponieważ współczynnik efektywności dla tego fragmentu planu jest niższy od progu weta równym 0,75. Po odblokowaniu węzła A (patrz rys. 3) podejmowane jest wykonanie oryginalnego planu z wykorzystaniem wyników uprzednio wykonanych fragmentów planu (oznaczonych odpowiednio X1 i X2).



Rys. 1. Opóźnienie danych z węzła A, znalezienie optymalnego fragmentu do realizacji



Rys. 2. Przedłużające się opóźnienie, znalezienie kolejnego fragmentu do wykonania



Rys. 3. Odblokowanie węzła A przetwarzanie jest kontynuowane z wykorzystaniem zmaterializowanych wyników pośrednich

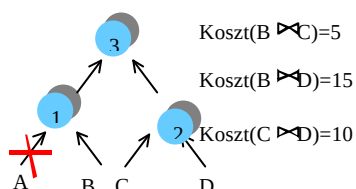
2.1. SYNTEZA OPERATORÓW

Algorytm *Query Scrambling* przechodzi do realizacji fazy syntezy operatorów jeżeli nadal istnieją zablokowane fragmenty planu zapytania i zostały wyczerpane wszystkie możliwości zmiany kolejności wykonania operatorów oryginalnego planu zapytania. W fazie syntezy operatorów optymalizator generuje nowy plan z uwzględnieniem faktu tymczasowego zablokowania części źródeł danych. W zależności od sposobu uwzględnienia zablokowania źródeł opracowano trzy strategie realizacji syntezy operatorów: *Pair*, *Include Delay* (ID), *Estimate Delay* (ED).

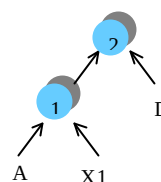
W strategii *Pair* optymalizator generuje wszystkie pary źródeł, które należy ze sobą połączyć, a następnie uwzględniając koszt realizacji tego połączenia i koszt materializacji, wykonywany jest operator połączenia z najniższym kosztem. Po zakończeniu realizacji wybranego uprzednio operatora optymalizator sprawdza dostępność zablokowanego węzła. Jeżeli węzeł ten pozostaje w stanie zablokowanym, to optymalizator wybiera kolejną parę źródeł danych do połączenia. Natomiast, jeżeli węzeł ten zostanie odblokowany, to optymalizator generuje całosciowy plan z uwzględnieniem uprzednio zmaterializowanych wyników pośrednich i odblokowanego źródła danych.

Na rysunku 4 zamieszczono przykładowy plan zapytania, który został przekazany z pierwszej fazy algorytmu *Query Scrambling* do drugiej fazy realizowanej za pomocą stra-

tegi *Pair*. Przetwarzanie zapytania nie może być kontynuowane z powodu zablokowania węzła A. Optymalizator szacuje koszt połączenia wszystkich par niezablokowanych źródeł. Koszty te zostały zamieszczone na rysunku. Optymalizator wybiera do połączenia parę o najniższym koszcie, czyli B i C. Po wykonaniu tej operacji pozostaje do połączenia A, D oraz X1 (wynik uprzedniego połączenia B i C). W następnym kroku optymalizator sprawdza dostępność węzła A. Jeżeli węzeł ten jest dostępny, to za pomocą klasycznej optymalizacji jest generowany plan łączący A, D oraz X1 (patrz rys. 5).



Rys. 4. Szacowanie kosztu połączenia wszystkich par niezablokowanych węzłów



Rys. 5. Plan zapytania po odblokowaniu węzła A

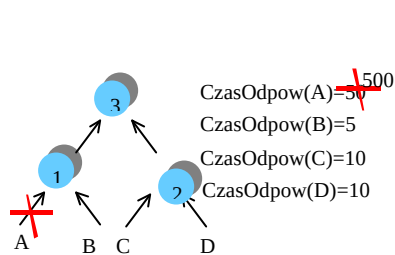
Strategie *Include Delay* oraz *Estimate Delay* w celu znalezienia nowego planu wykorzystują optymalizację opartą na czasie odpowiedzi. Optymalizacja taka umożliwia zamaskowanie opóźnienia spowodowanego zablokowaniem węzła. Strategie te zakładają określony czas opóźnienia zablokowanego węzła, a następnie wartość ta jest przekazywana do optymalizatora jako czas odpowiedzi tego źródła. W ten sposób dostęp do zablokowanych węzłów jest przesuwany w nowym planie wykonania zapytania na dalsze pozycje. W trakcie oczekiwania na dostęp do zablokowanych węzłów optymalizator może zlecić realizację innych fragmentów planu.

W strategii ID optymalizator przyjmuje arbitralnie określony czas opóźnienia zablokowanego węzła. Podstawowym problemem przy realizacji strategii ID jest odpowiednie określenie tego czasu. Oszacowanie zbyt długiego czasu może spowodować wybranie przez optymalizator kiepskiego planu. W celu zminimalizowania takiego niebezpieczeństwa wprowadza się współczynnik *ryzko-zysk*, który jest ilorazem ryzyka związanego z przyjęciem określonego czasu opóźnienia w stosunku do zysku jaki może przynieść podjęcie takiego ryzyka. Ryzyko definiuje się jako koszt wykonania fragmentu planu maskującego opóźnienie zablokowanego źródła danych. Zysk wylicza się jako różnicę między czasem realizacji oryginalnego planu z uwzględnieniem spodziewanego czasu opóźnienia i czasem realizacji planu wygenerowanego przez optymalizator z wykorzystaniem strategii ID.

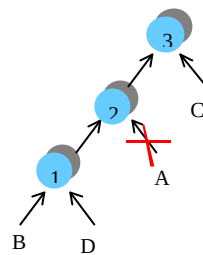
Współczynnik *ryzko-zysk* umożliwia sterowanie optymalizatorem w taki sposób aby unikał ryzykownych planów, które potencjalnie przynoszą znikomy zysk. Zwiększanie wartości tego współczynnika wymusza na optymalizatorze wybieranie bardziej konserwatywnych planów, które nie ponoszą nadmiernego ryzyka związanego ze zbyt pesymistycznym oszacowaniem opóźnienia.

Na rysunku 4 zamieszczono przykładowy plan zapytania przekazany do drugiej fazy wykonania realizowanej za pomocą strategii ID. Czasy odpowiedzi poszczególnych źródeł zostały umieszczone na rysunku. W wyniku zablokowania źródła A optymalizator arbitralnie określa czas odpowiedzi tego źródła na 500 i dokonuje optymalizacji opartej na czasie odpowiedzi uwzględniając współczynnik *ryzko-zysk* ustalany również arbitralnie. Wynik

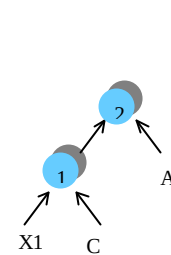
tej optymalizacji przedstawiony jest na rysunku 7. Jeżeli w czasie realizacji operatora 1 nastąpi, zgodnie z założeniami, odblokowanie źródła A, to wykonanie planu może być kontynuowane. Natomiast, jeżeli czas zablokowania przedłuży się ponad szacowany, to optymalizator przystępuje do generowania kolejnego planu w oparciu o strategię ID, według reguł opisanych powyżej. Przykładowy wynik takiej optymalizacji został przedstawiony na rysunku 8, gdzie X1 oznacza wynik działania operatora 1.



Rys. 6. Arbitralne określenie czasu odpowiedzi zablokowanego węzła



Rys. 7. Wygenerowanie nowego planu przy ustalonym czasie odpowiedzi zablokowanego węzła



Rys. 8. Odblokowanie węzła A przetwarzanie jest kontynuowane z wykorzystaniem zmaterializowanych wyników pośrednich

Strategia ED różni się od strategii ID jedynie sposobem szacowania czasu opóźnienia zablokowanego węzła. Strategia ED określa ten czas na 25% czasu wykonania całości oryginalnego planu i w miarę potrzeby sukcesywnie wydłuża go do 100%. Dopóki optymalizator obserwuje postęp w realizacji nowego planu, to szacowany czas opóźnienia pozostaje bez zmian. Jeżeli natomiast zostanie zaobserwowane zablokowanie tego planu, to optymalizator przyjmuje czas opóźnienia dwukrotnie dłuższy i dla takiego założenia generuje kolejny plan.

W opracowywaniu planu zapytania za pomocą strategii ED jest również wykorzystywany współczynnik ryzyko-zysk. Jednakże nie ma on tak dużego znaczenia jak w przypadku strategii ID, ponieważ strategia ED jest bardziej konserwatywna.

4. WYNIKI ESPERYMENTU SYMULACYJNEGO

W celu zbadania efektywności działania algorytmów *Query Scrambling* wykonano szereg eksperymentów symulacyjnych. Jako środowisko symulacyjne wykorzystano optymalizator kosztowy oparty na metaheurystykach *Simulated Annealing* i *Iterative Improvement* oraz zbiór testów TPC-D.

Z wyników eksperymentów można wyciągnąć następujące wnioski:

1. Efektywność wszystkich strategii zazwyczaj przewyższa efektywność klasycznej, statycznej optymalizacji. Zgodnie z oczekiwaniami algorytmy efektywnie wykorzystują czas opóźnienia węzła na wykonanie innych zadań. W zależności od fazy algorytmu czas ten jest wykorzystywany na przeszeregowanie operatorów lub opracowanie nowego planu.
2. Dla dużego bufora danych różnica między strategiami jest niewielka. W przypadku dużej wielkości pamięci operacyjnej koszt przechowywania zmaterializowanych wyników pośrednich jest znikomy i nie zaobserwowano znaczących różnic między poszczególnymi strategiami.

3. Dla malego bufora i przy bardzo małych opóźnieniach najlepszy jest oryginalny plan. W takich przypadkach oplaca sie przeczekac niedostepnosc wezla. Próba przeorganizowania planu powoduje porzucenie optymalnego planu i wybranie gorszego planu, który nie moze byc rekompensowany przez odpowiednio dlugo trwajace opóźnienie.
4. Dla malego bufora i przy stosunkowo małych opóźnieniach Strategie Pair i ED przewyższaja strategię ID. Strategia ID zaklada bardzo duze opóźnienia zablokowanego wezla, gdy oczekiwania mijaja sie z oczekiwaniami, to wygenerowany przez optymalizator plan daleki jest od optymalnego.
5. Dla malego bufora i przy srednich opóźnieniach ID i ED przewyższa Pair. W tym przypadku obserwuje sie wady strategii Pair, która nie generuje jednego calosciowego planu, lecz dziala krokowo wybierajac suboptymalne rozwiazania.
6. Przy duzych opóźnieniach strategie sa równowazne. W tym przypadku czas realizacji calego planu, nie zaleznie od przyjetej strategii, jest najsilniej zalezny od dlugosci opóźnienia. Bardzo dlugi czas opóźnienia jest w stanie zamaskowac dzialania wszystkich strategii.
7. Zmniejszajac współczynnik ryzyko-koszt efektywnosc strategii ID zbliża sie do strategii ED. Zmniejszenie tego współczynnika powoduje, ze strategia ID staje sie mniej agresywna i nie ponosi zbyt duzego ryzyka zwiazanego z arbitralnym przyjeciem dlugiego czasu zablokowania źródla.

Po analizie powyższych spostrzezen mozna przyjac, ze strategia, która posiada najlepsze wlasnoscí jest strategia ED.

5. PODSUMOWANIE

Prace badawcze, które zaowocowaly opracowaniem rodziny algorytmów *Query Scrambling*, stanowią pierwszy krok nad dynamicznymi, adaptacyjnymi algorytmami optymalizacji zapytan w srodowisku sieci rozleglej.

Dalsze prace beda sie wiazaly z wykorzystaniem replik danych dla zwiekszenia efektywnosci i niezawodnoscí w dostepie do danych w dynamicznym srodowisku przetwarzania. Odpowiednich badan wymaga również problem braku dostepnoscí danych statystycznych stanowiących podstawe dzialania współczesnych algorytmów optymalizacji.

Odrebny problem stanowi opracowanie technik przetwarzania udostepniajacych czesciowe wyniki, których jakosc satysfakcjonuje uzytkownika.

Obsluga zapytan z nieprecyzyjnymi kryteriami poszukiwania jest kolejnym zadaniem badawczym. Realizacja takich zapytan wymaga przygotowania indywidualnego profilu kryteriów uzytkownika, które pozwola precyzyjnie skonstruowac zapytanie do bazy danych.

LITERATURA

- [1] AMSALEG L., FRANKLIN M.J., TOMASIC A., URHAN T., *Scrambling Query Plans to Cope With Unexpected Delays*, Proc. of the 4th Int. Conf. On Parallel and Distributed Information Systems, Miami Beach (Florida), 1996, IEEE Computer Society Press, pp.208-219
- [2] AMSALEG L., FRANKLIN M.J., TOMASIC A., *Dynamic query operator scheduling for wide-area remote access*, Journal of Distributed and Parallel Databases, 6(3), 1998, pp.3-15
- [3] URHAN T., FRANKLIN M. J., AMSALEG L., *Cost based query scrambling for initial delays*, Proc. of ACM-SIGMOD, Seattle, USA, 1998, pp.130-141