

RECENZJA

rozprawy doktorskiej mgr Anny Sasak-Okoń

„Spekulatywne zrównoleglenie jako technika wspomaganie przetwarzania
kwerend w relacyjnych bazach danych”

1. Tematyka rozprawy

Z mojego punktu widzenia, tj. z punktu widzenia specjalisty od baz danych, rozprawa jest poświęcona współdzieleniu wyników podzapytań między zapytaniami zawierającymi te podzapytania. Choć kandydatka w swojej rozprawie posługuje się dość osobliwą terminologią jak na problematykę baz danych, jednak sam rozwiązywany przez nią problem jest niezwykle istotny dla dziedziny systemów baz danych. W produkcyjnych bazach danych strumień napływających zapytań nie jest przypadkowy. Pojawiające się w nim zapytania są pochodną funkcjonalności i architektury aplikacji wysyłających te zapytania. Zapytania zwykle powtarzają się; dotyczą tych samych lub podobnych podzbiorów danych i/lub mają wspólne podwyrażenia (podzapytania). Od wielu lat celem rozmaitych grup badawczych jest wykorzystanie możliwości wynikających z tej nieprzypadkowości zawartości strumienia zapytań z produkcyjnej bazy danych. Wyniki dotychczasowych badań są niestety fragmentaryczne. Okazuje się bowiem, że współdzielenie wyników najprostszych podzapytań powoduje duże problemy z rozmaitymi aspektami działania systemu zarządzania bazą danych, a w szczególności z gospodarką pamięcią operacyjną. Opracowano i komercyjnie udostępniono (w Microsoft SQL Server) kooperacyjny pełny przegląd tabel (ang. *cooperative scan*) polegający na współdzieleniu stron wczytywanych do pamięci operacyjnej w trakcie pełnego przeglądu zawartości tabeli. W eksperymentalnej bazie danych MonetDB zaimplementowano *database scrambling*, czyli coś w rodzaju adaptacyjnego indeksu w postaci samo-sortującej się tabeli w miarę nadchodzących zapytań. Ta koncepcja nie jest jednak nadal dostępna w żadnym komercyjnym systemie zarządzania bazami danych.

Na tym tle koncepcja przedstawiona w rozprawie jest bardzo obiecująca. Owszem, dotyczy ona tylko niewielkiego podzbioru współdzielonych podzapytań, tzn. selekcji z projekcjami. W porównaniu z obecnym stanem badań w tej dziedzinie, jest to istotny postęp. Tak szerokiej klasy podzapytań nie umie współdzielić na razie nikt. Kandydatka zaatakowała więc bardzo trudny problem i znalazła jego rozwiązanie. Być może ma ono pewne wady i można by wiele w nim poprawić. Wynik został jednak pozytywnie zweryfikowany eksperymentalnie.

2. Treść rozprawy i uwagi merytoryczne

Rozprawa składa się z ośmiu rozdziałów. Pierwsze cztery z nich stanowią wprowadzenie i zawierają definicje niezbędnych pojęć. Rozdział piąty stanowi definicję problemu. Rozdział szósty to opis rozwiązania. Rozdział siódmy to raport z eksperymentalnej weryfikacji zaproponowanego rozwiązania oraz propozycje usprawnień. Rozdział ósmy stanowi podsumowanie rozprawy.

Nienumerowany wstęp zawiera informacje o celu pracy i jej strukturze. Już tu na stronie 12 bardzo chciałbym przeczytać, czy systemie możliwe jest wykonanie spekulatywne zapytania, którego wynik będzie potem bezużyteczny. W pewnym sensie wyjaśnia się to samej rozprawie, bowiem spekulatywnie wykonywane zapytania są tu bardzo proste.

Rozdział pierwszy to wprowadzenie w problematykę obliczeń równoległych. Zawiera niemal wyłącznie informacje o zagadnieniach równoległości, a nie są w nim omówione zagadnienia baz danych (może z wyjątkiem punktu 1.4.2). Właśnie w tym punkcie pojawia się pojęcie *bazy transakcyjnej*, którego autorka niestety nie zdefiniowała. Czy jest baza danych OLTP, czy też baza danych, w której zaimplementowano transakcje zgodnie z aksjomatami ACID? Czy jeszcze coś innego? Z treści pracy można by wnioskować, że chodzi o to pierwsze znaczenie. Brak tej definicji prowadzi jednak do pewnych nieporozumień.

Rozdział drugi to omówienie relacyjnych baz danych oraz SZBD SQLite, który posłuży autorce do przeprowadzenia eksperymentów. W tym rozdziale najlepiej widać, że rozprawa powstała poza środowiskiem bazodanowym. Widać też pewne niedociągnięcia wynikające z braku obycia z badaniami naukowymi i terminologią baz danych. Na stronie 39 autorka pisze o „zależnościach funkcyjnych pomiędzy tabelami”, podczas gdy zależności funkcyjne dotyczą kolumn w ramach pojedynczej relacji. Domyślam się, że autorka miała zapewne na myśli zależności inkluzyjne (klucz obcy-klucz główny). Taki błąd terminologiczny (rzeczowy?) nie powinien jednak znaleźć się w rozprawie doktorskiej. Podobne braki występują na stronie 41, gdzie wymieniono operatory algebry relacji. Autorka nie wspomniała o trzech ważnych operatorach, którymi są zmiana nazw kolumn, przecięcie relacji i Θ -złączenie. Dwa ostatnie są definiowalne za pomocą pozostałych, jednak bez tego pierwszego nie da się wyrazić wielu ważnych zapytań algebry relacji, np. rozmaitych samozłączeń. Osobliwe jest też wyliczenie wersji standardu SQL na stronie 44. Wygląda na to, jakby czas tego standardu zatrzymał się w 1999 roku, podczas gdy było potem jeszcze kilka wersji SQL, które ukazały się w latach 2003, 2006, 2008 i 2011. Chociaż rozmaite usterki w tym rozdziale nie mają wpływu na wartość badań kandydatki, jednak rozprawa doktorska nie powinna ich zawierać.

Rozdział trzeci to w założeniu wprowadzenie do zagadnień obliczeń równoległych w bazach danych. Podobnie jak w przypadku rozdziału drugiego i w rozdziale trzecim widać, że autorka rozprawy nie pracuje na co dzień nad bazami danych. Obliczenia równoległe w bazach danych opisała z punktu widzenia systemów równoległych do pewnego stopnia ignorując to, co się wydarzyło w ostatnich latach w dziedzinie baz danych. Nieprawdziwa jest np. uwaga ze strony 61, że „*badania równoległości w kontekście baz danych skupiają się przede wszystkim wokół modelu relacyjnego, który daje dobrą podstawę dla równoległości danych*”. To właśnie odejście od modelu relacyjnego na początku XXI wieku i rozkwit baz danych nurtu NOSQL pozwoliły na równoległość obliczeń absolutnie nieosiągalną w klasycznych relacyjnych bazach danych. Do pewnego stopnia sam model danych jest ortogonalny wobec modelu równoległości. Dla tej ostatniej ważna jest głównie możliwość fragmentacji pionowej, czy poziomej znanej pod modnym ostatnio terminem *sharding*. Podobnie nie można zgodzić się ze stwierdzeniem, że nierównomierne obciążenie uzależnione jest od *wydajności* algorytmów partycjonujących (strona 62). Nie do końca wiem, jak autorka rozumie wydajność, jednak w większości przypadków jakość algorytmów obliczeń równoległych na wielkich danych zależy głównie od miar zrównoważenia podziału na partycje (*shards*) a nie od tego, jak szybko jest podział obliczany. Im mniejsza jest nierównowaga (skośność, ang. *skew*), tym cały algorytm będzie działał szybciej.

Pośród różnych rodzajów podziału danych autorka wyraźnie preferuje fragmentację poziomą. Dyskusja różnych rodzajów rozpraszania danych około strony 66 jest zdecydowanie zbyt pobieżna.

W bazach danych stosuje się rozmaite sposoby rozpraszania: fragmentację pionową, fragmentację poziomą, replikację oraz rozmaite rozwiązania hybrydowe będące mieszaniną tych pierwszych trzech. Ta ostatnia grupa jest najczęściej stosowana w praktyce, ponieważ złożona rzeczywistość przetwarzania danych w przedsiębiorstwach nie da się obsłużyć za pomocą jednego czystego akademickiego modelu obliczeń. Uwaga na dole strony 67 o tym, że pewne rodzaje partycjonowania się nie sprawdzają świadczy o tym, że kandydatka nie do końca zdaje sobie sprawę z zależności między profilem użycia bazy danych a optymalną fizyczną organizacją danych. Inaczej należy projektować model fizyczny bazy OLTP a inaczej bazy analitycznej. Wszystko zależy od tego, do czego baza danych będzie używana. Nie ma sposobu organizacji danych, który jest najlepszy w każdej sytuacji.

Omówienie rodzajów równoległości w bazach danych też przedstawia wiele do życzenia. W szczególności przykładem równoległości *inter-query* jest kooperacyjny pełny przegląd tabel (ang. *cooperative scan*), por. Marcin Żukowski, Sándor Héman, Niels Nes, Peter Boncz „*Cooperative Scans: Dynamic Bandwidth Sharing in a DBMS*”, VLDB 2007. Kandydatka na pewno świetnie zna literaturę z zakresu przetwarzania równoległego, jednak nie z zakresu baz danych.

Na stronie 74 pojawia się informacja, że w pewnym algorytmie liczba porównań wynosi $\frac{1}{2} * n * (n+1)$, co daje złożoność obliczeniową $O(n^2)$. Złożoność $O(n^2)$ jest w ogóle dziwna. Dlaczego nie $O(n)$? Poza tym taka liczba porównań oznacza złożoność $O(n^2)$. Czy to jest jakaś oczywista omyłka pisarska, czy też kandydatka nie rozumie pojęcia złożoności obliczeniowej? Z kolei na stronie 75 znajduje się następane osobliwe sformułowanie „*problem doboru ich kolejności [złączeń] uważany jest za NP-trudny*”? On po prostu jest taki. Dlaczego w rozprawie znajduje się niejasne sformułowanie „*jest uważany za*”?

Dyskusja o algorytmach złączenia ze strony 76 też ma pewne braki. Otóż autorka porównuje złączenie haszowane (ang. *hash-join*) ze złączeniem przez scalanie (ang. *merge-join*). Nie wspomina jednak o kilku kluczowych charakterystykach obu algorytmów. Złączenie haszowane jest bowiem wrażliwe na niedoskonałości funkcji haszującej, które mogą spowodować znaczącą nierównomierność rozkładu liczby wierszy w kubekach i w efekcie niską wydajność algorytmu. Złączenie przez scalanie nie ma tej wady. Z drugiej strony dużą zaletą złączenia haszowanego jest jego zdolność do wykorzystania nierównomiernych rozmiarów złączanych tabel, np. jeśli jedna tabela jest mniejsza niż dostępna pamięć operacyjna, można ją w całości rozrzucić do kubeków pamięci i nie haszować już drugiej tabeli. Dalej, na stronie 81 przy omówieniu sortowania z rozrzucaniem zakresów, nie wspomniano o bardzo ważnym algorytmie TeraSort i całym paradygmacie obliczeniowym MapReduce.

Rozdział czwarty zawiera informacje o obliczeniach spekulatywnych. Drugi punkt tego rozdziału jest poświęcony zastosowaniu obliczeń spekulatywnych w bazach danych. W szczególności na stronie 95 znajdują się uwagi o protokole blokowania dwufazowego (ang. *two-phase locking, 2PL*). Autorka wspomina, że pozytywne efekty może mieć osłabienie jego silnej wersji (*Strict 2PL*) poprzez wcześniejsze zdejmowanie blokad. Owszem zwiększy to współbieżność, jednak prowadzi do bardzo niekorzystnego zjawiska wycofań kaskadowych (ang. *cascading rollback*). Z powodu tego fenomenu, słabsza wersja 2PL nie jest stosowana w praktyce.

Rozdział piąty to jasne i zwięzłe postawienie celu i tezy pracy. Tak jak wspomniałem na wstępie niniejszej recenzji, uważam, że koncepcja badań opisanych w rozprawie jest niezwykle interesująca.

Rozdział szósty zawiera opis głównego wyniku badań kandydatki, mianowicie jednofazowego algorytmu wyboru wykonywanych kwerend spekulatywnych. Chociaż

zaproponowany algorytm jest stosunkowo prosty, wyniki eksperymentów (opisane w rozdziale siódmym) pokazują, że jest on skuteczny i osiąga bardzo dobre przyspieszenia. Mam do metody kandydatki kilka zasadniczych uwag.

Po pierwsze, dlaczego tylko jedno zapytanie spekulatywne może być wykorzystane w zapytaniu niespekulatywnym? Czy gdyby wykonywano złączenie dwóch zbiorów danych, które mamy spekulatywnie już obliczone, nie warto byłoby z nich skorzystać? Metoda opisana w rozprawie to wyklucza. Oczywiście mogłoby się okazać, że taka poprawka algorytmu powoduje pogorszenie jego jakości. W końcu mamy do czynienia li tylko z heurystyką. Jestem bardzo ciekaw, jakie byłyby efekty wykorzystania wielu zapytań spekulatywnych w jednym zapytaniu niespekulatywnym. Czy udałoby się to kandydatce sprawdzić przed publiczną dyskusją nad jej rozprawą?

Po drugie, uważam, że sposób generowania zapytań spekulatywnych z grafu zapytań jest bardzo nieefektywny. Prowadzi bowiem do wykładniczej liczby zapytań w oknie spekulacji, co zauważono w pracy. Wyobraźmy sobie jedną kolumnę, dla której w grafie spekulacji jest n krawędzi predykatów. Metoda kandydatki prowadzi do wygenerowania 2^n zapytań. Uważam, że można by tu zastosować technikę podobną do *database scrambling* (nie znam polskiego terminu), czyli na podstawie tych predykatów wygenerować maksymalnie $n + 1$ rozłącznych przedziałów i dla nich stworzyć zapytania spekulatywne. Następnie przy wyliczaniu zapytań niespekulatywnych należy korzystać z sum mnogościowych tych przedziałów. W ten sposób zamiast złożoności wykładniczej, otrzymamy złożoność liniową.

Po trzecie, uważam, że metoda kandydatki nie powinna uwzględniać modyfikacji danych. Przedstawione rozwiązanie zawiera bardzo prosty mechanizm reakcji na modyfikację danych. Jest on tak prosty, że wydaje się niemal bezużyteczny. Gdyby chcieć zrobić to bardziej subtelnie, trzeba byłoby zbudować tablice haszowane z kluczami wierszy z wyników zapytań spekulatywnych, a same zapytania musiałyby zachowywać klucz. To znacznie podniosłoby złożoność metody, ale także pozwoliło uniknąć wielu niepotrzebnych unieważnień wyników spekulacji. Z drugiej strony eksperymenty są prowadzone na benchmarku TPC-H, który jest przeznaczony dla analitycznych baz danych, gdzie modyfikacja jest bardzo rzadka. Po cóż więc mechanizm, który niewiele daje?

Na stronie 108 autorka zleca SZBD dbanie o trafność odpowiedzi. Zupełnie nie rozumiem, o co chodzi.

Występują też liczne nieścisłości, co do definicji wykorzystywanych struktur pomocniczych i dopuszczalnych konstrukcji w obsługiwanych zapytaniach. Na stronie 111 jest napisane, że zapytania zagnieżdżone zawsze zwracają dokładnie jedną wartość. Z drugiej strony dopuszczalne jest zagnieżdżanie zapytań w operatorze IN, który to operator ma sens tylko dla zbiorów wartości. Z kolei rysunek 6.3 na stronie 117 zawiera podzapytanie zwracające więcej niż jeden wiersz. I jeszcze cytat ze strony 155 „zgodnie ze zdefiniowaną w Podrozdziale 6.2 strukturą obsługiwanych zapytań, kwerenda zagnieżdżona w klauzuli WHERE zwraca zawsze pojedynczą wartość”. Takich sprzeczności nie powinno być.

Na stronie 113 jest mowa o krawędziach predykatów. Skąd się je bierze? Jak je wydedukować z modelu danych? Czy może one występują tylko w grafie zapytania? Ten problem widać też w pierwszym akapicie na stronie 114, gdzie jest mowa o ostatecznej postaci tego grafu. Definicja jest niejasna. Z kolei na przykładowym grafie te krawędzie raczej odpowiadają parom klucz obcy-klucz główny. Na stronie 116 brakuje wyjaśnienia, co się dzieje z kolumną, która jest i w SELECT i WHERE. Czy powstanie wtedy multigraf? Jak do tej pory autorka posługuje się jedynie zwykłymi grafami. Na stronie 119 autorka zauważa już, że taka sytuacja mogła nastąpić. Z drugiej strony multigrafy miały powstawać w wyniku analizy grup zapytań, a nie pojedynczych zapytań.

Na stronie 149 autorka opisuje sposoby szacowania wielkości wyniku zapytań. Te sposoby są u kandydatki niezwykle proste. Już od bardzo wielu lat do szacowania wielkości selekcji stosuje się histogramy. Metoda z pracy jest wyjątkowo prymitywna. Co więcej autorka nie wspomina o histogramach.

Rozdział siódmy zawiera raport z eksperymentalnej weryfikacji metody opisanej w rozprawie oraz pewne jej udoskonalenia. Wyniki testów dowodzą, że zaproponowana metoda jest skuteczna i istotnie poprawia wydajność baz danych. Z drugiej strony mam kluczowe zastrzeżenia do samej koncepcji testów. Autorka wielokrotnie pisze, że interesują ją sklepy internetowe, czyli bazy OLTP, natomiast do testów wykorzystuje benchmark TPC-H, który jest przeznaczony dla baz analitycznych i hurtowni danych. To jest duże nieporozumienie. Podobnie jak nazwanie na stronie 168 tabeli LINEITEM tabelą produktów. To jest główna tabela faktów tego benchmarku i są to pozycje zamówień a nie produkty. Oto cytat ze strony 168 „*Na Rysunku 7.1 zaprezentowano Schemat testowej bazy danych, która została wykorzystana do badań. Składa się ona z 8 tabel, tworząc model połączeń charakterystyczny dla sklepów internetowych*”. To jakieś nieporozumienie. To jest schemat hurtowni danych!

Ostatnia uwaga do tego rozdziału to potwierdzenie mojej drugiej zasadniczej uwagi do metody (tzn. do wykładniczego generowania zapytań kandydujących). Na stronie 173 w punkcie 7.4.1 pokazano liczby wygenerowanych zapytań spekulatywnych. Te liczby są ogromne, a przecież są generowane z kilku zapytań w oknie spekulacji. To pokazuje, że sama metoda generacji zapytań kandydujących może napotkać na nieprzewidywalne trudności przy zastosowaniach praktycznych.

Rozdział ósmy to podsumowanie wyników pracy. Mimo zastrzeżeń zaznaczam, że metoda zaproponowana przez kandydatkę okazała się skuteczna. Problem naukowy został rozwiązany.

3. Strona formalna rozprawy

Praca jest napisana dość przystępnym językiem. Można by pewnie istotnie poprawić ją przygotowując ją za pomocą LaTeXa a nie Worda. Wzory matematyczne znacznie ładniej wtedy wyglądają. W rozprawie jest pewne liczba drobnych błędów, także ortograficznych, które pozwolę sobie tu omówić.

Na stronie 11 jest „*ze występujące zależności danych*”, a powinno być „*ze względu na zależności danych*”.

Na stronie 12 jest „*wyniki wykonanych Kwerend Spekulatywne*”, a powinno być „*wyniki wykonanych Kwerend Spekulatywnych*”.

Na stronie 13 użyto skrótu CQAC, a wyjaśniono go dopiero na stronie 108.

Na stronie 14 jest „*średniej wielości sklep*”, a powinno być „*średniej wielkości sklep*”.

Na stronie 22 napisano „*mniej lub bardziej optymalnych*”. Nie jest jednak możliwe stopniowanie przymiotnika *optymalny*, tzn. najlepszy.

Na stronie 30 pojawia się termin *mutex*. Nie jestem zwolennikiem tej kalki z angielskiego. W polskiej terminologii przyjęło się mówić o semaforze, ew. semaforze binarnym.

Na stronie 49 baza danych SQLite jest nazwana *zagnieżdżoną* relacyjna baza danych. Jest to bardzo dyskusyjne tłumaczenie terminu *embedded*. Lepsze byłoby „*wbudowana*” lub „*biblioteczna*”.

Na stronie 64 jest „*liczy procesorów rzędu 30-40*”, a powinno być „*liczby procesorów rzędu 30-40*”.

Na stronie 72 znajduje się rysunek 3.9, którego zawartość nie jest przetłumaczona, mimo że w tekście używane są jedynie terminy polskie. Nie podano źródła pochodzenia tego rysunku.

Na stronie 80 pojawia się pojęcie *host*. Nie zdefiniowano go. Co to jest w tym kontekście?

Na stronie 83 jest mowa o „unii rezultatów”. Po polsku raczej mówimy o sumie (ew. teoriomnogościowej).

Na stronie 95 jest „*Impoving OLTP*”, a powinno być „*Improving OLTP*”.

Na stronie 101 w podpisie pod rysunkiem 5.1 słowo „*konkurencyjne*” (ang. *concurrent*) to po polsku raczej „*współbieżne*”.

Nie rozumiem następującego zadania ze strony 107: „*W odniesieniu do poprawności i aktualności danych w bazie danych, związanych ze spełnieniem aksjomatów ACID, używać się będzie określenia Stan Bazy Danych*”.

Na stronie 117 na rysunku 6.3 kolumna *name* z głównego zapytania jest podczepiona pod tabelę *Director* podczas, gdy główny SELECT odwołuje się tylko do tabeli *Movie*.

Na stronie 130 ten sam symbol *degS* użyty na oznaczenie dwóch różnych wartości: Spekulatywnego Stopnia Wierzchołka i Stopnia Selekcji Wierzchołka.

Na stronie 135 jest „*Spekulaywne*”, a powinno być „*Spekulatywne*”.

Następujące zdanie ze strony 159 jest niezrozumiałe: „*Wyniki zweryfikowane pozytywnie, o ile nie został przekroczony dopuszczalny rozmiar Bazy Wyników Spekulacji muszą być poddawane weryfikacji*”. Dlaczego zweryfikowane wyniki muszą być poddane weryfikacji?

4. Wniosek końcowy

Podsumowując, uważam, że w rozprawie rozwiązano ciekawy i bardzo ważny z praktycznego punktu widzenia problem naukowy. Eksperymentalna weryfikacja wyników badań nie pozostawia wątpliwości co do ich wartości naukowej. Stawiam hipotezę, że zastosowawszy pewne modyfikacje opisanej metody, można by osiągnąć wyniki lepsze. Trudno jednak to ocenić, ponieważ mamy tu do czynienia z heurystykami. Być może byłby to dobry kierunek dalszych badań? A może kandydatce do czasu publicznej dyskusji nad jej rozprawą uda się zweryfikować niektóre z tych hipotez?

Mam wiele zastrzeżeń do wprowadzających rozdziałów tej rozprawy i zwracam uwagę na pewne braki w wiedzy kandydatki z zakresu baz danych. Z drugiej strony, mimo pracy poza środowiskiem badaczy baz danych, udało się autorce uzyskać ciekawe wyniki z dziedziny baz danych. To bardzo dobrze świadczy o jej umiejętnościach badawczych i dojrzałości naukowej.

Wnioskuje o dopuszczenie pani mgr Anny Sasak-Okoń do dalszych etapów przewodu doktorskiego.



/-/ K.Stencel