STUDIA INFORMATICA 2012
Volume 33 Number 2B (106)

Michał LUPA, Adam PIÓRKOWSKI
Akademia Górniczo-Hutnicza w Krakowie, Katedra Geoinformatyki i Informatyki Stosowanej

REGUŁOWA OPTYMALIZACJA ZAPYTAŃ W BAZACH DANYCH PRZESTRZENNYCH

Streszczenie. Niniejszy artykuł porusza temat optymalizacji zapytań w bazach danych przestrzennych. Autorzy, jako główny nurt rozważań, przyjęli optymalizację regułową, która pozwala przyśpieszyć wykonywanie zapytań na etapie ich tworzenia. Efektem badań są spostrzeżenia w postaci proponowanych reguł optymalizacji, których efektywność została sprawdzona eksperymentalnie.

Zaproponowano trzy metody dekompozycji zapytań do postaci mniej czasochłonnych obliczeniowo. Pierwszą z nich jest zastąpienie funkcji przestrzennych w warunkach wyszukiwania operatorami logicznymi. W drugim przypadku do dekompozycji wykorzystano własność łączności w kontekście funkcji sumy. Trzecie rozpoznanie dotyczy własności łączności w aspekcie funkcji części wspólnej.

Słowa kluczowe: optymalizacja zapytań, regułowa optymalizacja zapytań, bazy danych przestrzennych, GIS

RULE-BASED QUERY OPTIMIZATION IN SPATIAL DATABASES

Summary. This article addresses how to optimize queries in spatial databases. Authors as mainstream considerations adopted to optimize adjustable speed performance which allows queries on the stage of their development. The result of the study are the observations in the form of proposed rules of optimization, where the effectiveness is verified experimentally.

Proposed three methods for decomposition of queries into a less time-consuming computationally. The first is to replace spatial functions in a Boolean search. In the second case, the decomposition property of communication used in the context of the function sum. The third property concerns the identification of communication in terms of joint function.

Keywords: query optimization, rule-based query optimization, spatial databases

1. Wprowadzenie

Na przełomie dekad odnotowano stale wzrastającą liczbę zastosowań relacyjnych baz danych wraz z rozszerzeniami dla danych przestrzennych (bazami danych przestrzennych) [1]. Niestety, w większości przypadków owe systemy służą jedynie jako magazyn dla takich danych, bowiem przetwarzanie tych danych najczęściej ma miejsce w specjalistycznych programach. Rozwój metod analizy danych przestrzennych po stronie bazy zaowocował standardami rozszerzeń języka SQL, a mianowicie pierwszym standardem według OGC [2, 3], dodającym podstawowe operacje na punktach i kształtach, a następnie drugim, czyli wydzieloną sekcją standardu SQL/MM [4], dotyczącą danych przestrzennych (SQL/MM – Spatial). Należy się spodziewać wzrostu zainteresowania także przetwarzaniem po stronie systemów zarządzania bazami danych. Wyprzedzając ów trend, skupiono się na problemach optymalizacji zapytań związanych z danymi przestrzennymi, albowiem temat ten jest rzadko poruszany, podczas gdy optymalizacja zapytań w ujęciu ogólnym jest zagadnieniem dobrze znanym [5, 6].

Wśród prac związanych z optymalizacją zapytań warto zwrócić uwagę na jedną z pierwszych prac [7] związanych z przekształceniami algebraicznymi zapytań. Niewątpliwie bardzo ciekawymi badaniami są analizy i propozycje optymalizacji złączeń na podstawie atrybutów przestrzennych [8, 9].

W pracach [10, 11] zwrócono uwagę na możliwość wykorzystania algebry Peano do dekompozycji zapytań. Osiągnięto przy tym wielokrotne skrócenie czasów wykonywania zapytań.

Innym podejściem pozwalającym na skrócenie czasu wykonywania zapytania jest generalizacja obiektów [12], która w przypadku generalizacji bezstratnej jest w pełni uzasadniona i efektywna, jeśli jest możliwa, a w przypadku generalizacji stratnej – przy zadanym wskaźniku jakości – pozwala na znaczne przyśpieszenie zapytań.

Warto wspomnieć także o aspekcie optymalizacji, jakim jest odpowiednia indeksacja. Dla danych przestrzennych wykorzystywane są specjalne konstrukcje indeksów. Temat ten został podjęty w pracy [13].

2. Możliwości optymalizacji zapytań przestrzennych

Przestrzenne rozszerzenia systemów zarządzania bazami danych wymagają dedykowanego podejścia do zagadnienia optymalizacji, ponieważ metody klasyczne nie gwarantują uzyskania satysfakcjonującego przyśpieszenia. Jednym z takich podejść jest zmiana sposobu konstruowania zapytań, które bazują na funkcjach przetwarzających dane geometryczne. Poniżej przedstawiono metody optymalizacji zapytań, które będą rozważane w niniejszej pracy:

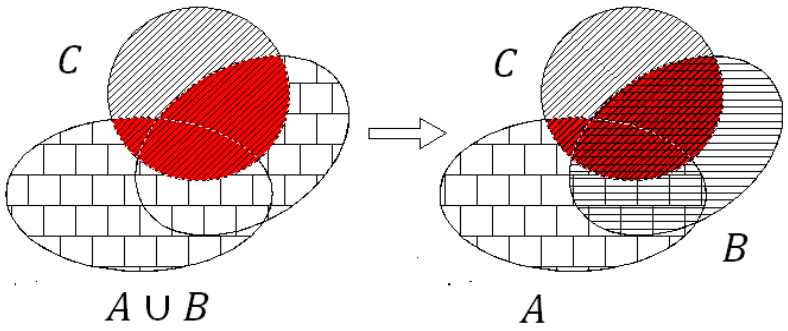
- zamiana funkcji przestrzennych na operatory logiczne,
- własność łączności w aspekcie funkcji sumy,
- własność łączności w aspekcie funkcji części wspólnej.

2.1. Zamiana funkcji przestrzennych na operatory logiczne

Zapytania uwzględniające złączenia danych geometrycznych z trzech lub więcej tabel są bardzo czasochłonne, a w przypadku zastosowania kombinacji funkcji przestrzennych w warunku złączenia można zaobserwować, iż czas potrzebny na zwrócenie wyników relatywnie wzrasta. Zaimplementowane narzędzia geoprocesingu (przetwarzania danych przestrzennych), takie jak suma i różnica, niekoniecznie są rozwiązaniem czasowo korzystnym, wręcz przeciwnie, często wydłużają, nawet kilkunastokrotnie, czas potrzebny do wykonania danego zapytania.

Pierwsza z proponowanych metod optymalizacji przedstawia możliwości zastąpienia przez operatory logiczne funkcji przestrzennych. Przekształcenia składni języka SQL oraz wykorzystanie słowa kluczowego „OR" powodują, iż możliwe jest skonstruowanie zapytania wyznaczającego sumę poligonów bez konieczności użycia narzędzia „UNION", alokującego pamięć na wyniki pośrednie. Pozwala to uniknąć stosowania funkcji zagnieżdżonych w warunkach złączenia zapytań dla trzech lub więcej warstw.

Przykładowym problemem, którego rozwiązanie można zoptymalizować za pomocą powyższych zależności, jest „przecięcie" trzech warstw, gdzie dwie z nich tworzą poligon powstały jako wynik sumy logicznej. Schemat ilustrujący powyższe zagadnienie przedstawiono na rys. 1.



Rys. 1. Przecięcie wielokątów w dwóch różnych wariantach
Fig. 1. The intersection of polygons in two different variants

W ramach badań wykorzystano przykładowe dane przestrzenne regionu Alaska, udostępnione przez twórców oprogramowania QuantumGIS [14]. Analiza ograniczyła się do trzech warstw o różnej wielkości oraz stopniu zróżnicowania geometrii: „tundra", „swamp" oraz „landice".

Zaproponowano następujące zapytania przestrzenne:

- Z1 – Przecięcie sumy warstw „tundra” i „swamp” utworzonej z wykorzystaniem funkcji ST_UNION oraz warstwy „landice” (zgodne z OpenGIS OGC i SQL/MM):

```
SELECT COUNT(*) FROM tundra, swamp, landice
WHERE
ST_INTERSECTS (ST_UNION(tundra.the_geom, swamp.the_geom), landice.the_geom);
```

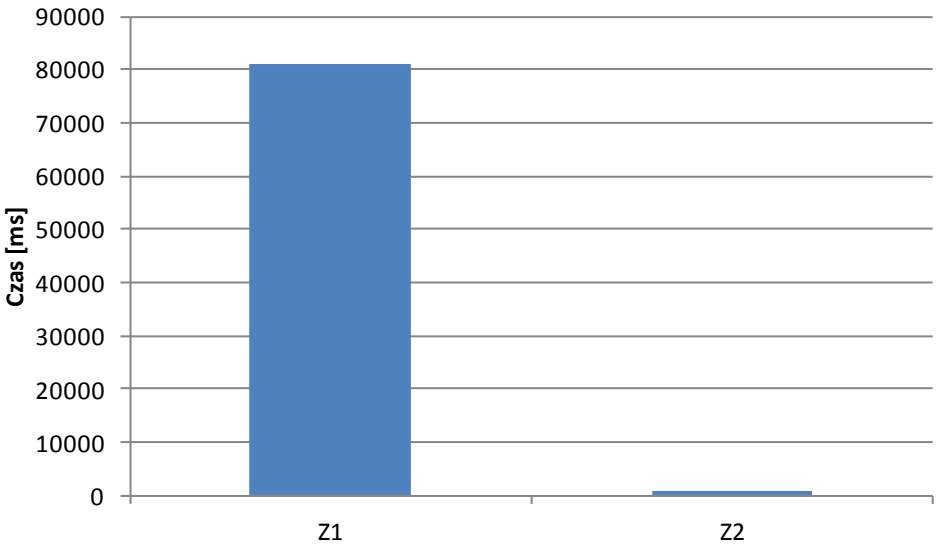
- Z2 – Zapytanie równoważne do Z1, przecięcie warstw „tundra” i „landice” lub „swamp” i „landice” skonstruowane za pomocą operatora logicznego „OR” (zgodne z SQL/MM):

```
SELECT COUNT(*) FROM tundra, swamp, landice
WHERE
ST_INTERSECTS(tundra.the_geom, landice.the_geom)
OR
ST_INTERSECTS(swamp.the_geom, landice.the_geom);
```

Tabela 1

Czasy zapytań Z1 i Z2

nr testu	Z1 [ms]	Z2 [ms]
1	81000	984
2	84485	937
3	81453	938



Rys. 2. Minimalne czasy wykonania zapytań Z1 i Z2
Fig. 2. Minimal query times for Z1 and Z2 queries

Zapytania wykonano na komputerze PC, który miał charakter stacji roboczej. Ponadto, wykorzystano system zarządzania bazą danych PostgreSQL 9.0.4 [15] wraz z rozszerzeniem przestrzennym PostGIS 1.5.0 [16]. System MySQL z wbudowaną opcją Spatial nie był brany pod uwagę z racji niekompletnej implementacji funkcji analitycznych [17]. Komputer posiadał procesor Intel Pentium IV 2,8 GHZ (Hyper-threading), 1GB RAM DDR1 333 MHz, HDD

Seagate 7200 obr/min. Powyższą konfigurację sprzętową oraz programową wykorzystano do przeprowadzenia wszystkich testów zawartych w niniejszej pracy.

Ze względu na długi czas wykonywania poszczególnych zapytań, zmniejszono liczbę rekordów w każdej tabeli o około 50%. Testy przeprowadzono trzykrotnie, wyniki przedstawiono w tabeli 1 oraz na wykresie (rys. 2).

Wyniki przeprowadzonych testów świadczą o tym, iż zapytania skonstruowane za pomocą operatora logicznego „OR" wykonują się ponad 85 razy szybciej (81000 ms dla „ST_UNION" oraz 934 ms dla „OR") niż te, w których zastosowano funkcję „ST_UNION". Potwierdza to słuszność omawianej metody, która pozwala na wielokrotne zwiększenie wydajności zapytań dotyczących analizy uwzględniającej przecięcia kilku warstw.

2.2. Własność łączności w aspekcie funkcji sumy

Kolejna z metod przedstawia możliwości strojenia zapytań przestrzennych realizujących sumę trzech lub więcej warstw. Wykorzystując algebrę zbiorów, można zauważyć, iż w przypadku sumy uogólnionej (tj. większej liczby zbiorów, w tym przypadku trzech obiektów geometrycznych) oraz jej własności – łączności – kolejność, w której podaje się składniki, ma znaczenie.

Własność łączności można zdefiniować w następujący sposób:

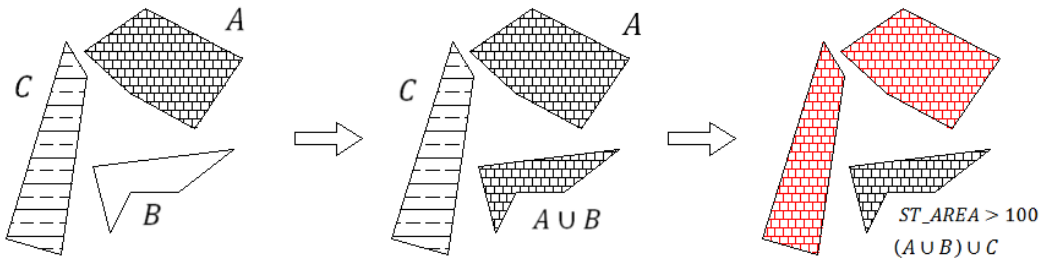
Dla dowolnych zbiorów A, B, C zachodzi następująca równość (1):

$$(A \cup B) \cup C = A \cup (B \cup C). \tag{1}$$

Powyższe równanie można przekształcić do postaci równoważnej, zapisanej za pomocą funkcji przestrzennych (2):

$$\text{ST\_UNION(ST\_UNION(A,B),C) = ST\_UNION(A,ST\_UNION(B,C))}. \tag{2}$$

Własność łączności zobrazowano na rys. 3. Na potrzeby testów przygotowano odpowiednie warstwy w celu lepszego zobrazowania wyników. Wykorzystano częściowe dane regionu Alaska [14]: „tundra" oraz „landice" (odpowiednio 80 i 30 krotek). Stworzono również poligon pomocniczy o zasięgu obu powyższych warstw, wielkości jednego wiersza.



Rys. 3. Graficzne przedstawienie zapytania testującego własność łączności sumy
Fig. 3. Graphical representation of the test queries the sum of property

Testy obejmowały trzy warianty zapytania uwzględniającego operację sumy, które w wyniku zwracało poligony o polu powierzchni większym niż 100 (ST_AREA > 100).

Zaproponowano trzy warianty zapytania testowego:

- Z3 – warstwa „zasieg" o najmniejszej wielkości i zróżnicowaniu (1 wiersz), jako składnik sumy zewnętrznej:

```
SELECT COUNT(*), FROM zasieg, tundra, landice
WHERE
ST_AREA(
ST_UNION(
ST_UNION(landice.the_geom, tundra.the_geom), zasieg.the_geom))>100;
```

- Z4 – warstwa „tundra" o największej wielkości o zróżnicowaniu (80 wierszy), jako składnik sumy zewnętrznej:

```
SELECT COUNT(*), FROM zasieg, tundra, landice
WHERE
ST_AREA(
ST_UNION(
ST_UNION(landice.the_geom, zasieg.the_geom),tundra.the_geom))>100;
```

- Z5 – warstwy „zasieg" oraz „tundra" jako składniki sumy wewnętrznej:

```
SELECT COUNT(*), FROM zasieg, tundra, landice
WHERE
ST_AREA(
ST_UNION(
ST_UNION(zasieg.the_geom,tundra.the_geom ),landice.the_geom))>100;
```

Wyniki testów zostały zebrane w tabeli 2, a następnie przedstawiono je graficznie (Rys. 4), uwzględniając tylko najkrótsze czasy dla poszczególnych zapytań.

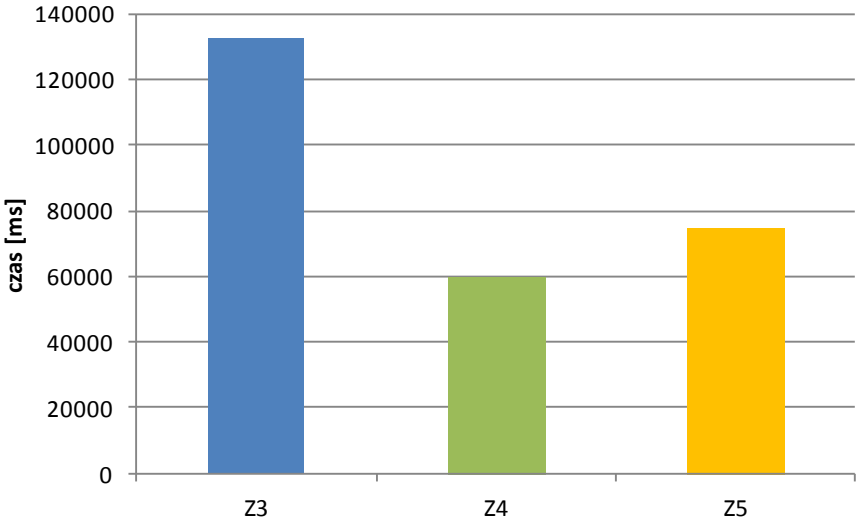
Tabela 2

Czasy wykonania zapytań Z3, Z4 i Z5 [ms]

nr testu	Z3 [ms]	Z4 [ms]	Z5 [ms]
1	132719	59640	75141
2	141091	60500	74844
3	159790	60244	75125

Przeprowadzone badania pokazują, iż suma, jako działanie algebry zbiorów oraz jej własność – łączność, może zostać wykorzystana w procesie optymalizacji zapytań przestrzennych. Na podstawie wykonanych testów, zawierających różne warianty zapytania skonstruowanego na podstawie własności łączności, można stwierdzić, iż kluczową rolę pełni kolejność poszczególnych elementów. Zapytanie (Z3), w którym warstwa o najmniejszym zróżnicowaniu i wielkości („zasieg") była składnikiem funkcji sumy zewnętrznej, wykonało się odpowiednio ok. 55% oraz 44% wolniej (132 719 ms) niż w przypadku, kiedy była ona składnikiem sumy wewnętrznej (Z4 oraz Z5). Najlepsze czasy (59 640 ms) uzyskano dla za-

pytania Z4, gdzie składnikiem sumy zewnętrznej była warstwa o największym zróżnicowaniu („tundra").



Rys. 4. Minimalne czasy wykonania wariantów zapytań Z3, Z4 i Z5 przedstawiających własność łączności sumy

Fig. 4. Minimal query times for Z3, Z4 and Z5 queries

2.3. Własność łączności w aspekcie funkcji części wspólnej

Kolejnym działaniem algebry zbiorów, bezpośrednio wykorzystywanym podczas analiz przeprowadzanych za pomocą zapytań przestrzennych, jest iloczyn mnogościowy (część wspólna). Podobnie jak w przypadku sumy, wyróżniamy własność łączności, którą można zdefiniować w poniższy sposób:

Dla dowolnych zbiorów A, B, C zachodzi następująca równość (3):

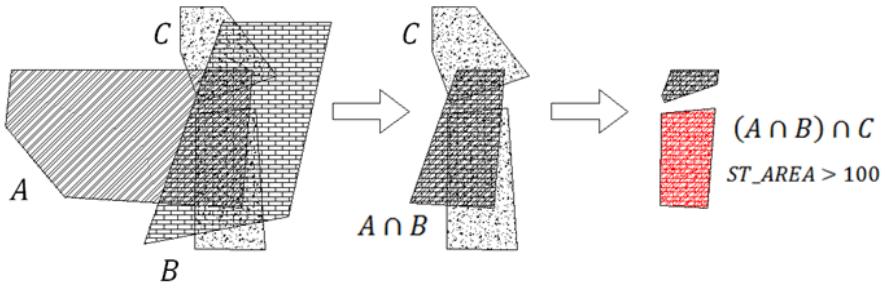
$$(A \cap B) \cap C = A \cap (B \cap C). \tag{3}$$

Postać równoważna powyższego równania, zapisana z wykorzystaniem funkcji przestrzennych (4):

ST_INTERSECTION(ST_INTERSECTION(A,B), C)

= ST_INTERSECTION(A, ST_INTERSECTION(B,C)). (4)

Przedstawiono, analogiczne do przypadku testów łączności dla funkcji sumy, zapytanie, które w wyniku zwracało część wspólną trzech warstw o polu powierzchni większym niż 100 (ST_AREA > 100).



Rys. 5. Graficzne przedstawienie zapytania testującego własność łączności części wspólnej
Fig. 5. Graphical representation of the test queries for intersection

Zaproponowano trzy warianty zapytania testowego:

- Z6 – warstwa „zasieg" o najmniejszej wielkości i zróżnicowaniu (1 wiersz), jako składnik iloczynu zewnętrznego:

```
SELECT COUNT(*), FROM zasieg, tundra, landice
WHERE
ST_AREA(
ST_INTERSECTION(
ST_INTERSECTION(landice.the_geom, tundra.the_geom), zasieg.the_geom))>100;
```

- Z7 – warstwa „tundra" o największej wielkości o zróżnicowaniu (80 wierszy), jako składnik iloczynu zewnętrznego:

```
SELECT COUNT(*), FROM zasieg, tundra, landice
WHERE
ST_AREA(
ST_INTERSECTION(
ST_INTERSECTION(landice.the_geom,
zasieg.the_geom),tundra.the_geom))>100;
```

- Z8 – warstwy „zasieg" oraz „tundra" jako składniki iloczynu wewnętrznego:

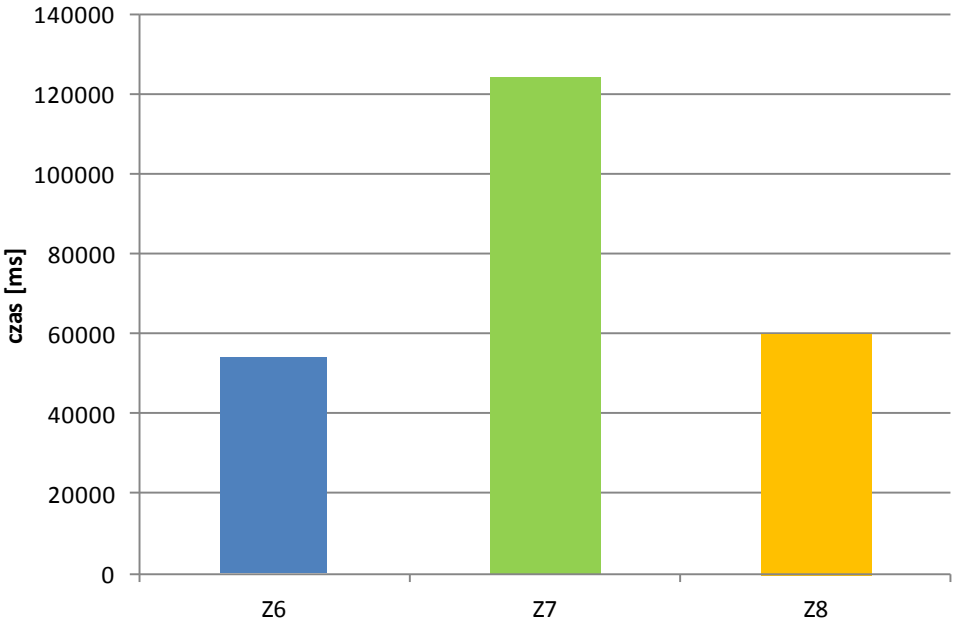
```
SELECT COUNT(*), FROM zasieg, tundra, landice
WHERE
ST_AREA(
ST_INTERSECTION(
ST_INTERSECTION(zasieg.the_geom,tundra.the_geom),landice.the_geom))>100;
```

Wyniki zebrano w tabeli 3 oraz przedstawiono graficznie najkrótsze czasy wykonania zapytań na wykresie (rys. 6).

Tabela 3

Czasy wykonania zapytań Z6, Z7 i Z8 [ms]

nr testu	Z6 [ms]	Z7 [ms]	Z8 [ms]
1	54734	124359	62031
2	53984	124422	60157
3	53875	124406	59953



Rys. 6. Minimalne czasy wykonania zapytań Z6, Z7 i Z8
Fig. 6. Minimal query times for Z6, Z7 and Z8 queries

Otrzymane wyniki pozwalają stwierdzić, iż łączność, w przypadku funkcji części wspólnej, jest własnością, która może być również zaadaptowana jako jedna z metod optymalizacji w kontekście zapytań przestrzennych baz danych. Analogicznie do sumy uogólnionej, kolejność, w której występują poszczególne warstwy, pełni tutaj kluczową rolę. Ponownie największą różnicę otrzymaliśmy w przypadku porównania funkcji zewnętrznej, której składnikami były warstwy o najmniejszym i największym zróżnicowaniu (Z6: „zasieg" oraz Z7: „tundra"). Jednakże w tym wypadku, najwolniej wykonało się zapytanie w wariancie Z7 (124359 ms), gdzie składnikiem iloczynu zewnętrznego była warstwa „tundra". Był to czas gorszy o ok. 57% od zapytania Z6 (53875 ms) oraz o ok. 52% (59953 ms) od Z8, w którym składnikiem funkcji zewnętrznej była warstwa „landice".

3. Wnioski

Reasumując, przeprowadzone badania pozwoliły wyróżnić propozycje reguł do metod optymalizacji zapytań przestrzennych:

- zastosowanie operatora logicznego „OR" jako zamiennika funkcji przestrzennej „ST_UNION" w zapytaniach badających przecięcie kilku warstw pozwala na uzyskanie przyśpieszenia rzędu kilkudziesięciu razy – wynika to m.in. z faktu, iż funkcja UNION tworzy dla każdego zapytania dodatkowe obiekty w pamięci,

- łączność jest własnością algebry zbiorów, którą można wykorzystać w procesie optymalizacji zapytań przestrzennych,
- w przypadku strojenia zapytań uwzględniających łączność w aspekcie sumy uogólnionej obiektów geometrycznych, warstwę o największym zróżnicowaniu należy uwzględnić jako składnik sumy zewnętrznej,
- użycie powyższej metody w celu optymalizacji zapytań wykorzystujących własność łączności dla części wspólnej da jednak odwrotny efekt. Przyśpieszenie uzyskano w momencie, gdy warstwa o najmniejszym zróżnicowaniu była składnikiem iloczynu zewnętrznego.

BIBLIOGRAFIA

1. Krawczyk A.: Próba systematyki zapisu atrybutów i topologii obiektów geometrycznych w systemach informacji geograficznej. Studia Informatica, Vol. 32, No. 2B (97), Wydawnictwo Politechniki Śląskiej, Gliwice 2011, s. 189÷201.
2. OGC – The Open Geospatial Consortium, http://www.opengeospatial.org/.
3. OpenGIS Implementation Specification for Geographic information – Simple feature access – SQL option. http://www.opengeospatial.org/standards/sfs.
4. ISO/IEC 13249-3:1999, Information technology – Database languages – SQL Multimedia and Application Packages – Part 3: Spatial, International Organization For Standardization, 2000.
5. Gurry M.: Optymalizacja Oracle SQL. Leksykon kieszonkowy. Helion, 2009.
6. Kostrzewa D., Josiński H.: Ocena jakości strategii eksploracji przestrzeni poszukiwań dla problemu określenia kolejności realizacji złączeń. Studia Informatica, Vol. 32, No. 2A (96), Wydawnictwo Politechniki Śląskiej, Gliwice 2011.
7. Helm R., Marriott K., Odersky M.: Constraint-Based Query Optimization for Spatial Databases. Proc. 10th ACM PODS, 1991.
8. Park H. H, Lee C. G., Lee Y. J., Chung. C. W.: Separation of Filter and Refinement Steps in Spatial Query Optimization. Technical Report CS/TR-98-122. Korea Advanced Instutute of Science and Technology.
9. Park H. H, Lee Y. J., Chung. C. W.: Spatial Query Optimization Utilizing Early Separated Filter and Refinement Strategy. Information Systems, Vol. 25, No. 1, 2000, s. 1÷22.
10. Bajerski P.: Optimization of geofield queries. Proceedings of the 1st International Conference on Information Technology, Gdańsk, Poland 2008.

11. Bajerski P., Kozielski S.: Computational Model for Efficient Processing of Geofield Queries. Proceedings of the International Conferencje on Man-Machine Interactions, Kocierz, Poland 2009.
12. Piórkowski A., Krawczyk A.: Wpływ generalizacji obiektów na optymalizację zapytań w bazach danych przestrzennych. Studia Informatica, Vol. 32, No. 2B(97), Wydawnictwo Politechniki Śląskiej, Gliwice 2011, s. 119÷129.
13. Gorawski M., Goławski K.: Indeks szkicu bazujący na aRB-drzewie, [w:] Kozielski S., Małysiak B., Mrozek D. (red.): Bazy Danych – Nowe Technologie – Architektura, metody formalne i zaawansowana analiza danych (T. 1), WKŁ, 2007.
14. Alaska –, http://download.osgeo.org/qgis/data/qgis_sample_data.zip.
15. PostgreSQL Home Page, http://www.postgresql.org/.
16. PostGIS Home Page, http://postgis.refractions.net/.
17. Piórkowski A.: Mysql Spatial And Postgis – Implementations of Spatial Data Standards. EJPAU 14(1), #03, 2011.

Wpłynęło do Redakcji 4 stycznia 2012 r.

Abstract

This article addresses how to optimize queries in spatial databases. Authors as mainstream considerations adopted to optimize adjustable speed performance which allows queries on the stage of their development. The result of the study are the observations in the form of proposed rules of optimization, where the effectiveness is verified experimentally.

Proposed three methods for decomposition of queries into a less time-consuming computationally. The first is to replace spatial functions in a Boolean search. In the second case, the decomposition property of communication used in the context of the function sum. The third property concerns the identification of communication in terms of joint function.

Adresy

Michał LUPA: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, al. Mickiewicza 30, 30-059, Kraków, Polska, mlupa@geol.agh.edu.pl.
Adam PIÓRKOWSKI: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, al. Mickiewicza 30, 30-059, Kraków, Polska, pioro@agh.edu.pl.