Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych

Szymon Górny

1. Wstęp

Zadanie polegało na analizie wpływu indeksowania na czas wykonywania zapytań, w systemach zarządzania bazami danych, dla tabel znormalizowanych i zdenormalizowanych. Analizę przeprowadzono korzystając z PostgreSQL i Microsoft SQL Server. Podczas wykonywania zadania wzorowano się na artykule „*Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych”*.

2. Konfiguracja sprzętowa i programowa

Testy przeprowadzono na komputerze o podanej specyfikacji:

- CPU: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 2801 MHz, Rdzenie: 4, Procesory logiczne: 8
- GPU: NVIDIA GeForce GTX 1060 with Max-Q Design, 6GB
- RAM: 16 GB, DDR4 2400MHz
- S.O: Windows 10 x64
- PostgreSQL 13.2
- SQL Server 15.0.2000.5

3. Kryteria testów

W teście wykonano szereg zapytań sprawdzających wydajność złączeń i zagnieżdżeń z tabelą geochronologiczną w wersji znormalizowanej i zdenormalizowanej. Procedurę testową przeprowadzono kolejno dla tabel z oraz bez nałożonych indeksów dla obu systemów zarządzania bazami danych.

Dodatkowo, utworzone zostały tabele pomocnicze:

- Dziesiec

```
CREATE TABLE Dziesiec(cyfra INT, bit INT);
INSERT INTO Dziesiec VALUES
    (0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1),(8,1),(9,1);
```

- Milion

```
CREATE TABLE Milion(liczba int,cyfra int, bit int);
INSERT INTO Milion SELECT a1.cyfra +10* a2.cyfra +100*a3.cyfra + 1000*a4.cyfra + 10000*a5.cyfra
+ 100000*a6.cyfra AS liczba, a1.cyfra AS cyfra, a1.bit AS bit
FROM Dziesiec a1, Dziesiec a2, Dziesiec a3, Dziesiec a4, Dziesiec a5, Dziesiec a6;
```

Zapytania, które zostały użyte do przeprowadzenia testów:

- 1ZL

```
SELECT COUNT(*) FROM Milion
INNER JOIN GeoTabela ON (mod(Milion.liczba,68)=(GeoTabela.id_pietro));
```

- 2ZL

```
SELECT COUNT(*) FROM Milion
INNER JOIN  GeoPietro  ON (mod(Milion.liczba,68)=GeoPietro.id_pietro)
NATURAL JOIN GeoEpoka
NATURAL JOIN GeoOkres
NATURAL JOIN GeoEra
NATURAL JOIN GeoEon;
```

- 3ZG

```
SELECT COUNT(*) FROM Milion
WHERE mod(Milion.liczba,68) = (SELECT id_pietro FROM GeoTabela WHERE mod(Milion.liczba,68)=(id_pietro));
```

- 4ZG

```
SELECT COUNT(*) FROM Milion
WHERE mod(Milion.liczba,68) IN (SELECT GeoPietro.id_pietro FROM GeoPietro
                                NATURAL JOIN GeoEpoka
                                NATURAL JOIN GeoOkres
                                NATURAL JOIN GeoEra
                                NATURAL JOIN GeoEon);
```

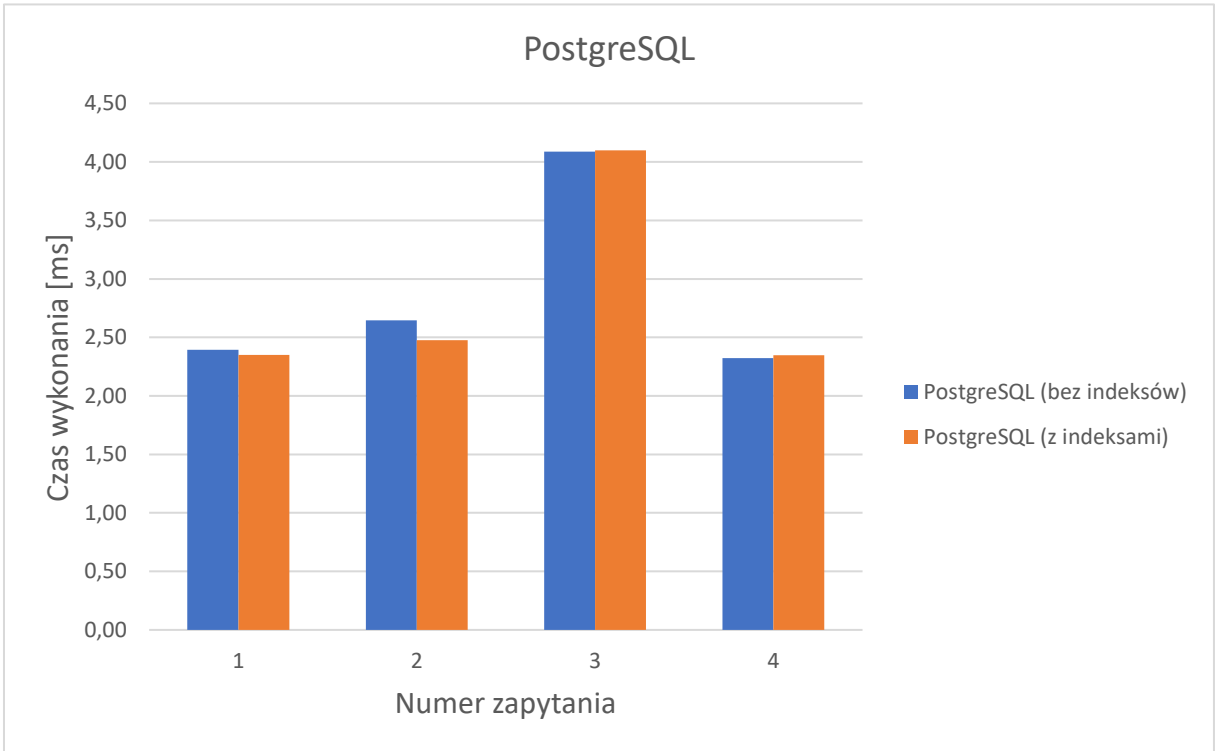
4. Wyniki

Podczas wykonywania analizy wywołano trzydzieści razy każde z zapytań - zarówno dla tabel z oraz bez nałożonych indeksów. Następnie uzyskane wyniki zostały uśrednione i zapisane w Tabeli 1. Arkusz ze szczegółowymi danymi pomiarów został umieszczony na repozytorium wraz z pozostałymi plikami.

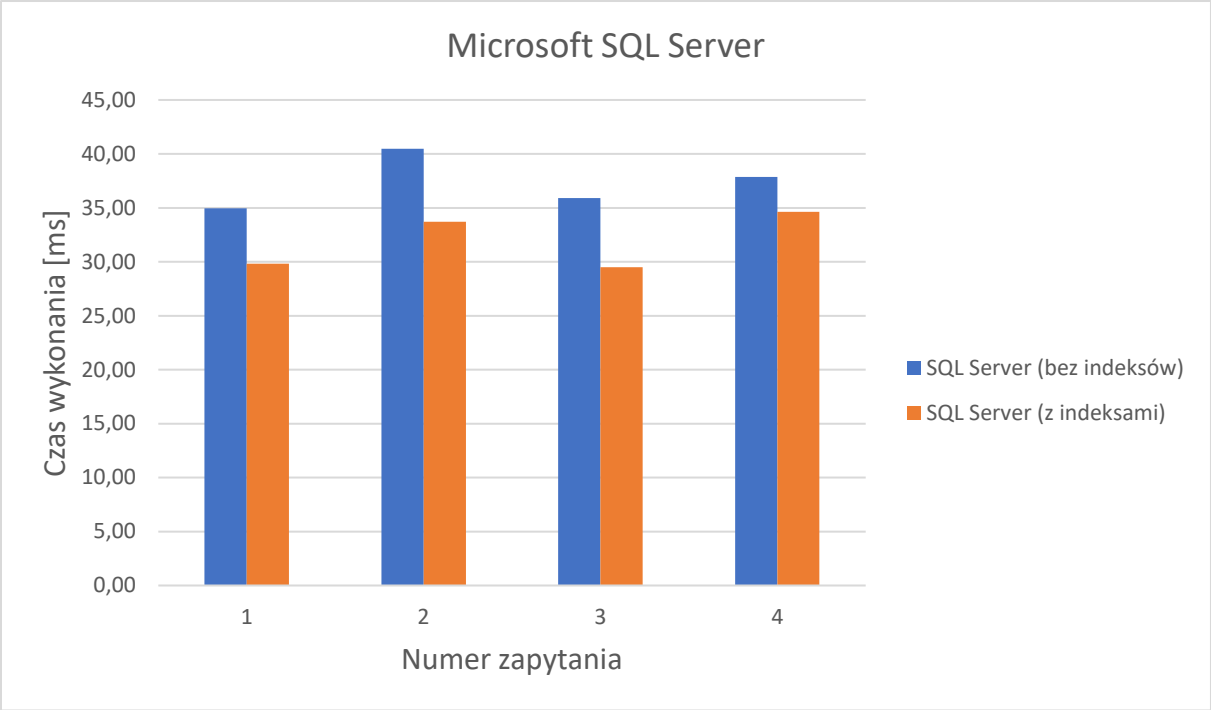
	1ZL	2ZL	3ZG	4ZG
PostgreSQL (bez indeksów)	247,63	441,57	12273,53	209,70
PostgreSQL (z indeksami)	223,53	298,80	12568,37	222,60
SQL Server (bez indeksów)	34,97	40,47	35,90	37,87
SQL Server (z indeksami)	29,83	33,70	29,50	34,63

Tabela 1. Średnie czasy wykonywania zapytań w milisekundach

Wyniki z Tabeli 1 zostały przedstawione na Rysunku 1 i Rysunku 2 w postaci histogramu. Dla Rysunku 1 przeskalowano oś pionową na logarytmiczną, dla lepszej wizualizacji danych.



Rysunek 1. Średnie czasy wykonywania zapytań w milisekundach z osią pionową przeskalowaną na logarytmiczną



Rysunek 2. Średnie czasy wykonywania zapytań w milisekundach

5. Wnioski

Analizując otrzymane wyniki, przy użyciu PostgreSQL, można zauważyć, że indeksacja przyspieszyła proces wykonywania zapytań tylko dla połowy przypadków (10,8% dla 1ZL oraz 47,8% dla 2ZL). W pozostałych przypadkach czas wykonywania po indeksacji uległ niewielkiemu zwiększeniu (2,3% dla 3ZG oraz 5,8% dla 4ZG), co niekoniecznie źle świadczy o samej indeksacji - wydłużenie czasu może wynikać na przykład z procesów działających w tle lub oprogramowania w którym wykonywano testy. Z kolei, wszystkie czasy wykonywania zapytań otrzymane w Microsoft SQL Server uległy zmniejszeniu po zastosowaniu indeksacji (17,2% dla 1ZL, 20,1% dla 2ZL, 21,7% dla 3ZG, 9,3% dla 4ZG). Porównując ze sobą wyniki z obu systemów zarządzania bazami danych, można zauważyć, że dla każdego zapytania zdecydowanie szybszy okazał się Microsoft SQL Server (przyspieszenie wyniosło co najmniej kilkaset procent w stosunku do PostgreSQL!).

Dla analizowanych danych, różnica w wykonywaniu zapytań przed i po indeksacji była prawie niezauważalna, bowiem była to kwestia kilkunastu/kilkudziesięciu milisekund, jednak w przypadku znacznie bardziej rozbudowanych baz danych, indeksacja może mieć znaczący wpływ na skrócenie czasu wykonywania zapytań.

6. Bibliografia

[1] Ł. Jajeśnica, A. Piórkowski, „*Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych*”