Wpływ indeksów na szybkość wykonania zapytań dla złączeń i zagnieżdżeń w schematach znormalizowanych i zdenormalizowanych

Długosz Hubert
06.06.2021r.

1. Cel ćwiczenia

Realizowane zadanie polegało na zbadaniu wpływu indeksowania jako mechanizmu wykorzystywanego w celu optymalizacji baz danych SQL dla złączeń i zagnieżdżeń w schematach znormalizowanych i zdenormalizowanych. Analizę przeprowadzono dla dwóch systemów zarządzania bazami danych SQL Server oraz PostgreSQL. Całość została oparta na artykule *„Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych”* autorstwa Łukasza Jaśnicy oraz Adama Piórkowskiego.

2. Metodyka

W celu wykonania analizy zbudowano trzy tabele dla których wykonywane były zapytania – tabela geochronologiczna zdenormalizowana, tabela geochronologiczna znormalizowana oraz tabela *Milion* wypełniona kolejnymi liczbami naturalnymi od 0 do 999 999. Każde zapytanie łączyło jedną z tabel przechowujących dane geochronologiczne z tabelą *Milion*.

- Zapytanie 1ZL, będące złączeniem syntetycznej tablicy miliona liczb z tabelą geochronologiczną zdenormalizowaną:

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON
(mod(Milion.liczba,77)=(GeoTabela.id_pietro));
```

- Zapytanie 2ZL, będące złączeniem syntetycznej tablicy miliona liczb z tabelą geochronologiczną znormalizowaną wymagające złączenia pięciu tabel:

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON
(mod(Milion.liczba,77)=GeoPietro.id_pietro) NATURAL JOIN GeoEpoka
NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

- Zapytanie 3ZG, będące złączeniem syntetycznej tablicy miliona liczb z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie jest wykonane przez zagnieżdżenie skorelowane:

```
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77) =
(SELECT id_pietro FROM GeoTabela WHERE mod(Milion.liczba,77)=(id_pietro));
```

- Zapytanie 4ZG, będące złączeniem syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest wykonane przez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem pięciu tabel:

```
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77) IN
(SELECT GeoPietro.id_pietro FROM GeoPietro NATURAL JOIN GeoEpoka
NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon);
```

Całość analizy została podzielona na dwa etapy dla każdego systemu zarządzania bazami danych. Pierwszy etap wykonywany był dla tabel bez nałożonych indeksów na kolumny danych. Dziesięć razy wywołano każde zapytanie i dla każdego z nich zapisywany był czas jego wykonania. Drugi etap zrealizowany był dla tabel z nałożonymi indeksami na wszystkie kolumny biorące udział w złączeniu. Czasy wykonania zapytań dla SQL Server znajdują się w Tabeli 1, natomiast dla PostgreSQL w Tabeli 2.

3. Konfiguracja sprzętowa

Analiza odbyła się na komputerze o następującej specyfikacji:

CPU: Intel Core i5-4570 3.20GHz
GPU: GTX 760
RAM: DDR3 8GB
SSD: 550/500 Mb/s
S.O: Win10 Pro x64
SQL Server 15.0.2000.5
PostgreSQL 13.2

Starano się, aby warunki podczas przeprowadzania analizy były niezmienne.

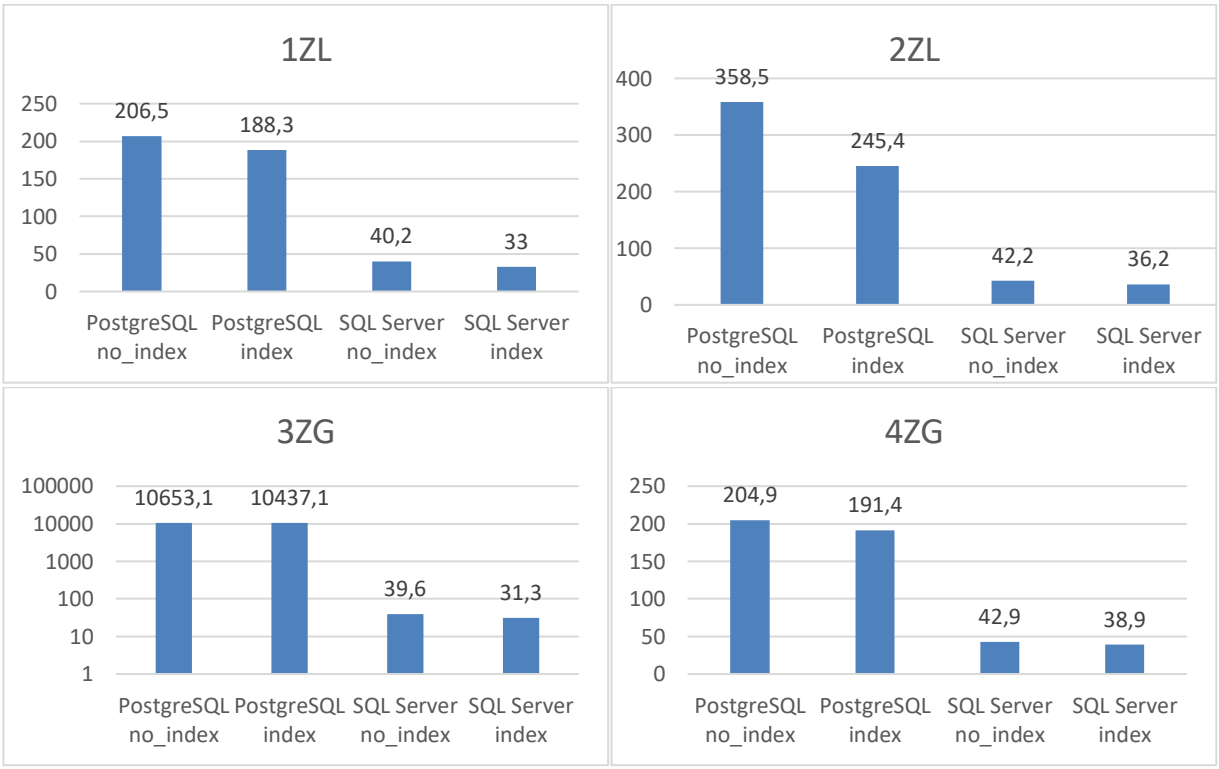
4. Wyniki:

	SQL Server							
	Bez indeksów				Z indeksami			
	1ZL D	2ZL N	3ZG D	4ZG N	1ZL D	2ZL N	3ZG D	4ZG N
1.	38	45	49	43	41	34	34	37
2.	38	40	37	43	24	38	27	39
3.	44	41	37	62	34	40	31	38
4.	38	42	42	42	36	32	34	37
5.	38	39	41	40	38	39	31	42
6.	47	46	38	37	28	36	38	39
7.	38	45	35	42	32	33	24	41
8.	41	41	40	38	32	35	37	40
9.	43	44	36	42	39	33	33	38
10.	37	39	41	40	26	42	24	38
Wartość średnia	40,2	42,2	39,6	42,9	33	36,2	31,3	38,9
Wartość minimalna	37	39	35	37	24	32	24	37

Tabela 1. Czasy wykonania zapytań w SQL Server w milisekundach

	PostgreSQL							
	Bez indeksów				Z indeksami			
	1ZL D	2ZL N	3ZG D	4ZG N	1ZL D	2ZL N	3ZG D	4ZG N
1.	231	331	10589	200	180	245	10143	185
2.	199	346	10629	197	184	250	10536	199
3.	203	377	10573	213	191	268	10589	190
4.	207	358	10618	195	187	243	10581	188
5.	205	338	10693	208	182	242	10170	200
6.	197	387	10670	210	204	254	10582	186
7.	205	365	10732	209	189	245	10549	196
8.	201	348	10677	203	196	251	10585	187
9.	198	350	10695	213	182	235	10061	184
10.	219	385	10655	201	188	231	10575	199
Wartość średnia	206,5	358,5	10653,1	204,9	188,3	246,4	10437,1	191,4
Wartość minimalna	197	331	10573	195	180	231	10061	184

Tabela 2. Czasy wykonania zapytań w PostgreSQL w milisekundach



Wykres 1. Średnie czasy wykonania poszczególnych zapytań w milisekundach (3ZG przedstawione w skali logarytmicznej)

Kody źródłowe, schematy planów zapytań oraz pozostałe wyniki znajdują się w repozytorium pod adresem https://github.com/hdlugosz/bazy-danych/tree/main/cw9.

5. Wnioski:

Bazując na otrzymanych wynikach można wysnuć wniosek, iż indeksacja jako pomocnicza struktura danych poprawia szybkość wykonywania kwerend SQL. Dla każdego z czterech zapytań, zarówno złączeń jak i zagnieżdżeń skorelowanych dla SQL Server, jak i dla PostgreSQL czas wykonania się zmniejszył. W najgorszym przypadku było to przyspieszenie rzędu 2% dla trzeciego zapytania w PostgreSQL. Największa poprawa wydajności stanowiła natomiast ponad 40% dla drugiego zapytania w tym samym systemie zarządzania bazami danych. Dla analizowanych danych była to często niemalże niezauważalna różnica rzędu kilku – kilkunastu milisekund, jednakże dla bardziej skomplikowanych oraz znacznie większych baz danych różnica ta mogłaby mieć znaczący i zauważalny wpływ. Dodatkowo patrząc na otrzymane wyniki, widoczna jest zdecydowana różnica w czasie egzekucji pomiędzy PostgreSQL oraz SQL Server na korzyść SQL Server.

6. Bibliografia:

Ł. Jajeśnica, A. Piórkowski, *Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych*