

Test wydajności Harmonogramu

Projekt: "Przeprowadzenie prac badawczych i rozwojowych umożliwiających wdrożenie inteligentnego kontraktu opartego o technologię blockchain"

Autorzy:

- Blicharski Bartłomiej
- Martin Morawiec

TimeSpan Postgres Extension

Ponieważ po stronie bazy danych PostgreSQL w wersji 11, zostało napisane rozszerzenie, które umożliwia użycie funkcji przeszukiwania harmonogramu, która z kolei uruchamia timespan. Kod odczytujący harmonogramy został napisany w języku Go lang, a następnie odpowiednio zmodyfikowany. W celach zbadania, czy dane rozwiązanie będzie odpowiednio wydajne przetestowano wykonanie odczytu z bazy danych już za pomocą funkcji, użytej podczas zapytania do bazy danych. A następnie wykonano serię testów ilościowych.

Użyty harmonogram:

```
{ "or": [ { "and": [ { "wd": [ 1 ] }, { "or": [ { "bhr": [ 10 ], "ehr": [ 12 ] }, { "bhr": [ 12,30 ], "ehr": [ 15 ] } ] } ] }, { "and": [ { "wd": [ 2, 3, 4, 5 ] }, { "or": [ { "bhr": [ 8 ], "ehr": [ 12 ] }, { "bhr": [ 12,30 ], "ehr": [ 15 ] } ] } ] }, { "wd": [ 6 ], "bhr": [ 8 ], "ehr": [ 12 ] } ] }
```

```
SELECT * FROM gtfs_routes WHERE IsOpen(ARRAY[2019,10,12],schedule::TEXT);
```

Ilość wierszy	
22	14 ms 14/22 = 0.64 ms / wiersz
815	281 ms 281/815 = 0.344 ms / wiersz
4115	1338 ms (1.34 s) 1388/4115 = 0.337 ms / wiersz

```
SELECT * FROM gtfs_routes WHERE IsOpen(ARRAY[2019,10,12],schedule::TEXT) = false;
```

Ilość wierszy	
4115	1184 ms (1.18 s) 1388/4115 = 0.337 ms / wiersz

TimeSpan v1.01

Optymalizacja kodu harmonogramu spowodowała wielokrotnie lepszy wynik wyszukiwań. W teście zastosowano 11 tysięcy rekordów tras (gdzie skopiowano 4 różne harmonogramy).

Przykładowe Dane:

```
INSERT INTO public.gtfs_routes VALUES (NULL, 1, 'Route A', 'Route A',
'Route A', 1, '', '', '', '', 10, '{"or": [{"and": [{"wd": [1]}, {"or":
[{"hr": [12], "bhr": [10]}, {"hr": [15], "bhr": [12, 30]}]}]}, {"and":
[{"wd": [2, 3, 4, 5]}, {"or": [{"hr": [12], "bhr": [8]}, {"hr": [15],
"bhr": [12, 30]}]}]}, {"hr": [12], "wd": [6], "bhr": [8]}]}' , NULL);
```

```
INSERT INTO public.gtfs_routes VALUES (NULL, 1, 'Route B', 'Route B',
'Route B', 1, '', '', '', '', 1, '{"and": [{"bhr": [12]}, {"or": [{"hr":
[2], "wd": [2]}, {"hr": [4], "wd": [4]}]}]}', NULL);
```

```
INSERT INTO public.gtfs_routes VALUES (NULL, 1, 'Route C', 'Route C',
'Route C', 1, '', '', '', '', 1, '{"and": [{"ev": [2019, 10, 10]}, {"and":
[{"not": [{"bev": [2019, 10, 3]}]}]}, {"or": [{"wd": [1], "bhr": [2]}, {"wd":
[3], "bhr": [4]}]}]}, {"or": [{"hr": [6], "wd": [5]}, {"hr": [8], "wd":
[7]}]}]}', NULL);
```

```
INSERT INTO public.gtfs_routes VALUES (NULL, 1, 'Route D', 'Route D',
'Route D', 1, '', '', '', '', 12, '{"or": [{"and": [{"wd": [1], "bev":
[2019, 10, 8]}, {"or": [{"hr": [12], "bhr": [10]}, {"hr": [15], "bhr": [12,
30]}]}]}, {"and": [{"wd": [2, 3, 4, 5]}, {"or": [{"hr": [12], "bhr": [8]},
{"hr": [15], "bhr": [12, 30]}]}]}, {"hr": [12], "wd": [6], "bhr": [8]}]}' ,
NULL);
```

```
SELECT * FROM gtfs_routes WHERE IsOpen(schedule::TEXT, ARRAY[2019,10,12]);
```

Ilość wierszy	
11204	1,353 ms (1.35 s) 1350/11204 = 0.12 ms / wiersz

With LIMIT 1000;

Ilość wierszy	
11204 (limit 1000)	118.81 ms (0.18 s) 0.18/1000 = 0.00018 ms / wiersz