

## Задание 2. Производительность индексов

1. На основе предложенного списка сгенерированы ~1000000 записей
2. Реализован функционал /user/search с поиском 'SELECT \* FROM "user" where first\_name like ? and second\_name like ? order by id'
3. Проведены тесты

Число тредов	Среднее время ответа до индексации	Среднее время ответа после индексации	Пропускная способность до индексации	Пропускная способность после индексации
1	28.79	7.81	33.34	118.12
10	65.85	12.59	140.27	592.07
100	636.07	92.74	147.28	730.99
500	3162.53	292.06	147.19	886.05
1000	6453.90	794.83	145.58	778.66

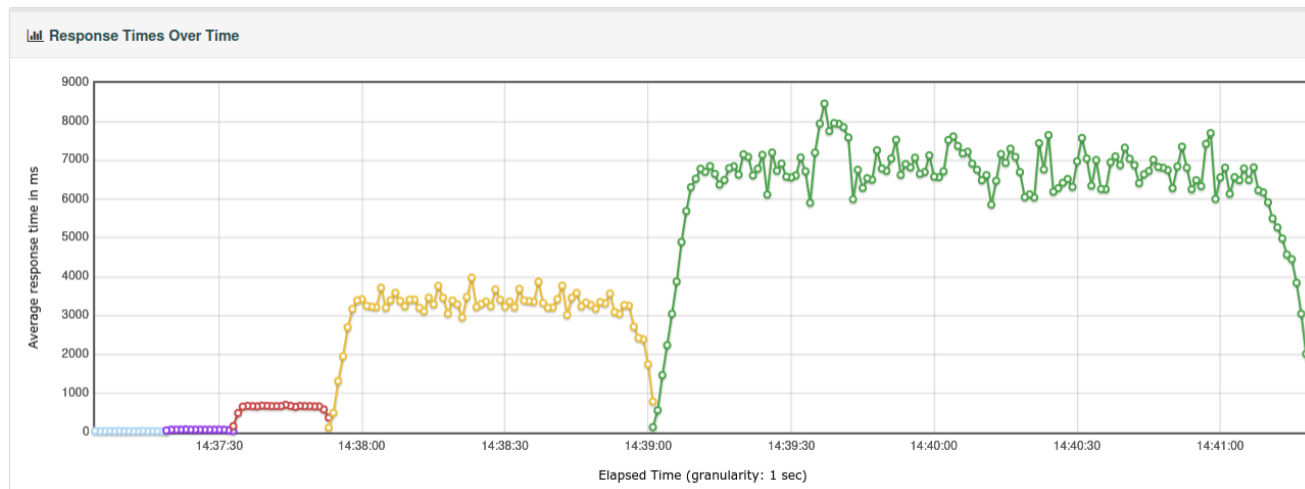
4. Анализ и графики до индексации

```
h1=# explain SELECT * FROM "user" where first_name like 'A6%' and  
second_name like 'Л%' order by id;
```

### QUERY PLAN

```
-----  
-----  
Sort  (cost=23334.69..23334.70 rows=1 width=129)  
  Sort Key: id  
    -> Gather  (cost=1000.00..23334.68 rows=1 width=129)  
        Workers Planned: 2  
          -> Parallel Seq Scan on "user"  (cost=0.00..22334.58 rows=1  
width=129)  
                Filter: (((first_name)::text ~~ 'A6% '::text) AND  
((second_name)::text ~~ 'Л% '::text))  
(6 rows)
```

## Время ответа



## Пропускная способность



### 5. Создан индекс

Предлагается использовать триграммный индекс, поскольку алгоритм его работы (разбиение на триграммы и далее маппинг триграмм на список строк, которые содержат триграмму) дает эффективное ограничение перебора по введенным для поиска символам.

```
h1=# CREATE EXTENSION IF NOT EXISTS pg_trgm;  
CREATE EXTENSION
```

```
h1=# CREATE INDEX idx_user_first_name_trgm ON "user" USING gin  
(first_name gin_trgm_ops);  
CREATE INDEX
```

```
h1=# CREATE INDEX idx_user_second_name_trgm ON "user" USING gin  
(second_name gin_trgm_ops);  
CREATE INDEX
```

## Анализ после индексации

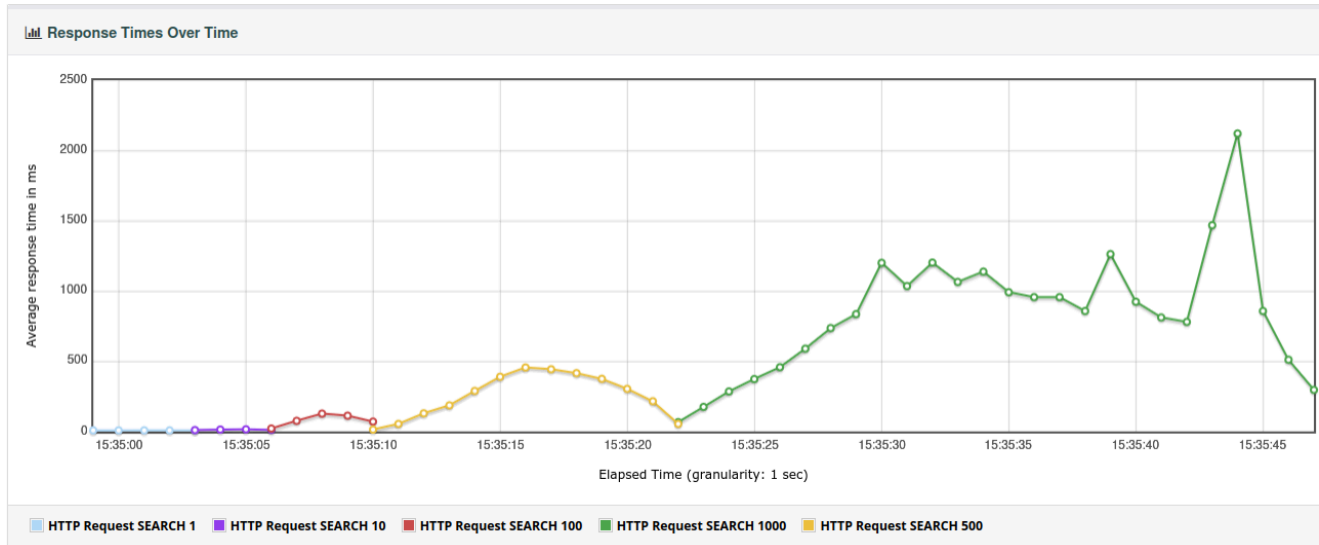
```
hl1=# explain SELECT * FROM "user" where first_name like 'A6%' and
second_name like 'Л%' order by id;
```

#### QUERY PLAN

```
-----
Sort  (cost=191.96..191.96 rows=1 width=101)
  Sort Key: id
    -> Bitmap Heap Scan on "user"  (cost=187.93..191.95 rows=1 width=101)
          Recheck Cond: (((first_name)::text ~~ 'A6% '::text) AND
((second_name)::text ~~ 'Л% '::text))
            -> BitmapAnd  (cost=187.93..187.93 rows=1 width=0)
                  -> Bitmap Index Scan on idx_user_first_name_trgm
                        (cost=0.00..30.05 rows=62 width=0)
                              Index Cond: ((first_name)::text ~~ 'A6% '::text)
                  -> Bitmap Index Scan on idx_user_second_name_trgm
                        (cost=0.00..157.63 rows=23217 width=0)
                              Index Cond: ((second_name)::text ~~ 'Л% '::text)
(9 rows)
```

## 6. Графики после индексации

### Время ответа



# Пропускная способность

