Сравнительный анализ полнотекстового поиска в Elasticsearch, postgreSQL и sphinx:

Сравнительные характеристики поиска при стемминге и сортировке по релевантности по запросу «большой театр»:

- **Sphinx – 128мс**
- **Elasticsearch – 428мс**
- **PostgreSQL – 43800мс**

```
import time
query =  "большой театр"
```

- **Elasticsearch**

```
from elasticsearch import Elasticsearch
es = Elasticsearch()
t = time.time()
res = es.search(index="items-index3",
                body={"query": {"simple_query_string": {
                    'query': query,
                    'fields': [ 'name^2', 'annotation' ]
          }}})
print("Найдено %d записей за %0.3fс:" % (res['hits']['total'],
time.time()-t))
for hit in res['hits']['hits'][:10]:
    print(hit["_id"],'\t',hit["_score"],'\t',hit["_source"]["name"])
```

Найдено 245381 записей за 0.428с:

```
25187953    55.79111   Большой театр
20467618    54.35475   Большой театр
17922732    53.739975  Наш Большой театр
2659641     52.498142  Трупы Большого театра
32876879    52.41837   Наш Большой театр
7113590     52.401215  Трупы Большого театра
3752910     52.294712  Художники Большого театра
2890717     52.153015  Большой Драматический Театр
19119731    51.889175  Большой театр "Малышка"
6132303     51.21914   Картина-гравюра большая "Большой театр"
```

- **PostgreSQL**

```
import psycopg2
```

```
conn = psycopg2.connect(dbname="postgres", host="localhost",
user="postgres", password="111")
cur = conn.cursor()
t = time.time()
cur.execute(
    "SELECT id, name, ts_rank(make_tsvector(name,annotation),q) as
rank"+
    " FROM items, plainto_tsquery('russian','"+query+"') AS q"+
    " WHERE make_tsvector(name,annotation) @@ q"+
    " ORDER BY ts_rank(make_tsvector(name,annotation),q) DESC"+
    " limit 10;")
print("Найдено за %0.3fc:" % (time.time()-t))
for id_,name, rank in cur.fetchall():
    print(id_,'\t',rank,'\t',name)
cur.close()
conn.close()
```

Найдено за 43.800c:

```
6132303     0.999995   Картина-гравюра большая "Большой театр"
27030530    0.999992   Оперная труппа Большого театра
3146998     0.999986   История русского балета. Мариинский театр. Большой
театр
3237308     0.999982   Большой театр (Москва). Пазл, 1000 элементов
24929395    0.999953   Большой Театр СССР. Опера. Балет
6077685     0.999933   Большой Театр СССР (комплект из 4 книг)
2488560     0.999929   Календарь настенный перекидной на 2006. Большой
театр
3562005     0.9999     Большой театр СССР. История сооружения и
реконструкции здания
5629128     0.999894   Большой театр СССР: опера, балет
19064032    0.999819   CubicFun "Большой театр", 29 элементов
```

- **Sphinxsearch**

```
from sphinxapi import *
cl = SphinxClient()
cl.SetServer ( '35.234.137.224', 9312 )
cl.SetMatchMode ( SPH_MATCH_ALL )
cl.SetLimits ( 0, 10 )
res = cl.Query ( query, 'test1' )
cl.SetSortMode(SPH_SORT_RELEVANCE)
print('Найдено {} записей за {}
c:'.format(res['total_found'],res['time']))
for rec in res['matches']:
    print(rec['id'],'\t',rec['weight'])
```

```
print(res['words'])
```

Найдено 3057 записей за 0.128с:

```
68332       4
73615       4
154978      4
1214580     4
1270954     4
1387794     4
2076490     4
2170711     4
2183610     4
2364686     4
[{'word': 'больш', 'docs': 283289, 'hits': 345960}, {'word': 'театр',
'docs': 19980, 'hits': 30828}]
```

Инсталляция и проверка работоспособности:

elasticsearch

Установка на 4-х нодах

- `wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -`
- `sudo apt-get install apt-transport-https`
- `echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list`
- `sudo apt-get update && sudo apt-get install elasticsearch`

Конфигурация /etc/elasticsearch/elasticsearch.yml на всех 4-х нодах

- `cluster.name`: *npl*,
- `node.name`: *instance1 - имя текущего хоста*,
- `network.host`: 0.0.0.0
- `http.port`: 9200
- `discovery.zen.ping.unicast.hosts`: ["instance2", "instance3", "instance4"] — имена хостов (все минус нода, на которой мы меняем конфиг).

Запуск сервиса на всех нодах (ставится в автозапуск)

```
sudo -i service elasticsearch start
```

Проверка успешного запуска

```
curl -XGET 'localhost:9200/?pretty'
curl -XGET 'localhost:9200/_cluster/health?pretty'
```

Создание русского анализатора в индексе

```
curl -XPUT -H 'Content-Type: application/json' "http://
35.195.166.173:9200/items-index" -d'
{
  "settings": {
    "analysis": {
      "filter": {
        "ru_stop": {
          "type": "stop",
          "stopwords": "_russian_"
        },
        "ru_stemmer": {
          "type": "stemmer",
          "language": "russian"
        }
      },
      "analyzer": {
        "default": {
          "char_filter": [
            "html_strip"
          ],
          "tokenizer": "standard",
          "filter": [
            "lowercase",
            "ru_stop",
            "ru_stemmer"
          ]
        }
      }
    }
  }
}'
```

Закачка данных в Elk (data2elk.py)

```
import json
import codecs
from elasticsearch import Elasticsearch
from elasticsearch import helpers

es = Elasticsearch()

actions = []
with codecs.open('data/item_details_full.txt',encoding='utf-8') as f:
  for i,line in enumerate(f):
    rec = json.loads(line)
    action = {
    "_index": "items-index",
    "_type": "items",
    "_id": int(rec["itemid"]),
    "_source":
{"name":rec.get("attr1",""),"annotation":rec.get("attr0","")}
    }
    actions.append(action)
    if (i+1) % 100000 == 0:
      print i
      helpers.bulk(es,actions)
```

```
      actions = []
if len(actions) > 0:
  helpers.bulk(es,actions)
print i
```

Тестирование

http://35.195.166.173:9200/items-index/_search?q=Спокойная%20ночь&pretty


```
{
  "took" : 3631,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 47843,
    "max_score" : 24.610964,
    "hits" : [
      {
        "_index" : "items-index",
        "_type" : "items",
        "_id" : "32979088",
        "_score" : 24.610964,
        "_source" : {
          "name" : "Спокойной ночи  ",
          "annotation" : "В красочную книгу петербургской
писательницы Дины Телевицкой вошли сказки и стихи для детей.   "
        }
      },
      {
        "_index" : "items-index",
        "_type" : "items",
        "_id" : "32598175",
        "_score" : 24.610964,
        "_source" : {
          "name" : "Спокойной ночи  ",
          "annotation" : "Картонная книжка-ширма \"Спокойной
ночи\". Текст книжки - это чувашская народная детская песенка. В
пересказе А.Лайко.  "
        }
      },
…
     {
        "_index" : "items-index3",
        "_type" : "items",
        "_id" : "82146",
        "_score" : 24.428816,
        "_source" : {
          "name" : "Спокойной ночи  ",
```

```
          "annotation" : "<GENERIC>Music Box Lullaby; Засыпающий
малыш; Где-то там, далеко  "
         }
       }
     ]
   }
}
```

Привязка к nginx (/var/www/search/index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Search demo site</title>
</head>
<body>
<h1>Elasticsearch client side demo</h1>
<div id="search_container">
    <label for"search">Search</label>
    <input type="text" id="search"/>
    <input type="submit"
onclick="doSearch(document.getElementById('search').value);"/>
</div>
<div id="total"></div>
<div id="hits"></div>
<script type="application/javascript">
  function doSearch (needle) {
    var searchHost = 'http://35.195.166.173:9200/items-index/items/
_search';
    var body = {
      'size': 10
    };
    if (needle.length !== 0) {
      var query = {
        'bool': {}
      };
      if (needle.length !== 0) {
        query.bool.must = {
          'simple_query_string': {
            'query': needle,
            'fields': [ 'name^2', 'annotation' ]
          }
        };
      }
      body.query = query;
    }

    var xmlHttp = new XMLHttpRequest();
    xmlHttp.open('POST', searchHost, false);
    xmlHttp.setRequestHeader('Content-Type', 'application/
json;charset=UTF-8');
    xmlHttp.send(JSON.stringify(body));
    var response = JSON.parse(xmlHttp.responseText);
```

```
    // Print results on screen.
    var output = '';
    for (var i = 0; i < response.hits.hits.length; i++) {
      output += '<h3>' + response.hits.hits[i]._source.name + '</h3>';
      output += response.hits.hits[i]._source.annotation + '</br>';
    }
    document.getElementById('total').innerHTML = '<h2>Showing ' +
response.hits.hits.length + ' results</h2>';
    document.getElementById('hits').innerHTML = output;
  }
  doSearch('');
</script>
</body>
</html>
```

postgreSQL

Установка

```
sudo apt install postgresql
```

Конфигурация (/etc/postgresql/9.5/main/pg_hba.conf)

```
local   all         postgres                          peer
# TYPE  DATABASE        USER            ADDRESS                 METHOD
# "local" is for Unix domain socket connections only
local   all   postgres                                peer
# IPv4 local connections:
host    all   postgres          all         mdc
# IPv6 local connections:
host    all   postgres          all           md5
```

Создание таблицы

```
CREATE TABLE items
(
    id integer NOT NULL,
    name text,
    annotation text,
    CONSTRAINT items_pkey PRIMARY KEY (id)
)
```

Загрузка данных

1)

```
postgres=# copy items from '/home/ubuntu/de3/lab1/postgres/data/
items.tsv' WITH DELIMITER E'\t';
```

2)

```
import psycopg2
conn = psycopg2.connect(dbname="postgres", host="35.195.166.173",
user="postgres", password="111")
cur, err = conn.cursor(), 0
with open('/home/ubuntu/de3/lab1/postgres/data/items.tsv', 'r') as f:
    #cur.copy_from(f, 'items', sep='\t')
    for i, line in enumerate(f):
        row = line.replace('\n','').split('\t')
        try:
            cur.execute(
                "INSERT INTO items VALUES (%s, %s, %s)",
                row
```

```
            )
        except Exception as Ex:
            err += 1
            print(i+1,j,str(Ex))
            print(row)
        if (i+1) % 100000 == 0:
            print(i+1,j)
            conn.commit()
conn.commit()
print(i+1,err)
```

Создание полнотестовой ф-ции по 2-м полям

```
CREATE OR REPLACE FUNCTION make_tsvector(title text, content text)
RETURNS tsvector AS $$
  BEGIN
    RETURN (setweight(to_tsvector('russian',title),'A')) ||
(setweight(to_tsvector('russian',content),'B'));
  END;
$$ LANGUAGE plpgsql IMMUTABLE;
```

Создание полнотекстового индекса

```
CREATE INDEX IF NOT EXISTS idx_fts_items on items
  USING gin(make_tsvector(name,annotation));
```

Проверка поиска

```
SELECT id, name
FROM items, plainto_tsquery('russian','путин выбор') AS q --
to_tsquery('russian','путин & выбор') AS q
WHERE make_tsvector(name,annotation) @@ q
ORDER BY ts_rank(make_tsvector(name,annotation),q) DESC
limit 10;
```

sphinx

Установка

https://www.8host.com/blog/ustanovka-i-nastrojka-sphinx-v-ubuntu-16-04/

Конфигурирование (/etc/sphinxsearch/sphinx.conf)

```
      sql_host        = 35.195.166.173
      sql_user        = postgres
      sql_pass        = *
      sql_db          = postgres
      sql_port        = 5432     # optional, default is 3306
      sql_query       = \
           SELECT id, name, annotation \
           FROM items
index test1
{
…
      morphology      = stem_en, stem_ru, soundex
…
}
```

Индексирование:

- index: sudo indexer --all
- reindex: sudo /usr/bin/indexer --rotate --config /etc/sphinxsearch/sphinx.conf --all

Запуск сервиса
start: sudo sed -i 's/START=no/START=yes/g' /etc/default/sphinxsearch
sudo /etc/init.d/sphinxsearch start

Тестирование индекса
run: mysql -h0 -P9306

```
mysql> SELECT * FROM test1 WHERE MATCH('путинов выбора'); SHOW META;
+----------+
| id       |
+----------+
|  3455568 |
|  7317705 |
|  7281469 |
| 28337620 |
| 32844123 |
|  8171861 |
| 19102396 |
|  3570652 |
| 20391242 |
|  4986214 |
| 20383587 |
| 24167237 |
|  7824583 |
|  8681407 |
| 18821192 |
| 19388064 |
| 20897626 |
| 24166016 |
| 28285156 |
| 28289226 |
+----------+
20 rows in set (0.01 sec)

+---------------+------------+
| Variable_name | Value      |
+---------------+------------+
| total         | 154        |
| total_found   | 154        |
| time          | 0.003      |
| keyword[0]    | путин      |
| docs[0]       | 1880       |
| hits[0]       | 3424       |
| keyword[1]    | выбор      |
| docs[1]       | 89311      |
| hits[1]       | 98873      |
+---------------+------------+
9 rows in set (0.00 sec)
```