Sphinx

Open Source Search Server

2013 . 01 . 03 Vilnius PHP

Mini Prisistatymas...

- Vaidas Žilionis
- vaidas@zilionis.net / +37061691393
- Skype: vaidas_zilionis
- Twitter: @zilionis
- http://www.zilionis.net (almost RIP)
- http://www.linkedin.com/in/vaidaszilionis

Iš akmens amžiaus prisiminimų

What is it?
You have new mail!
TOM

Iš akmens amžiaus prisiminimų

-
  ```
  SELECT `id`, `data` FROM `table` WHERE `data` like ('%puodukas%')
  ```

-
  ```
  SELECT `id`, `data` FROM `table` WHERE (`data` like ('%kavos%') AND `data` like ('%puodukas%'))
  ```

-
  ```
  SELECT
  `id`, `title`, `description`, `text` FROM `table`
  WHERE
  (
         (`title` like ('%kavos%') AND `title` like ('%puodukas%'))
   OR (`description` like ('%kavos%') AND `description` like ('%puodukas%'))
   OR (`text` like ('%kavos%') AND `text` like ('%puodukas%'))
  )
  ```

Iš akmens amžiaus prisiminimų

- SELECT `id`, `data` FROM `table` WHERE `data` like ('%puodukas%')

- SELECT `id`, `data` FROM `table` WHERE (`data` like ('%kavos%') AND `data` like ('%puodukas%'))

-
  ```
  SELECT
  `id`, `title`, `description`, `text` FROM `table`
  WHERE
  (
         (`title` like ('%kavos%') AND `title` like ('%puodukas%'))
   OR (`description` like ('%kavos%') AND `description` like ('%puodukas%'))
   OR (`text` like ('%kavos%') AND `text` like ('%puodukas%'))
  ```

Iš akmens amžiaus prisiminimų

```
SELECT
`id`, `title`, `description`, `text` FROM `table`
WHERE
(
        (`title` like ('%kavos%')
             AND `title` like ('%puodukas%'))
 OR (`description` like ('%kavos%')
             AND `description` like ('%puodukas%'))
 OR (`text` like ('%kavos%')
            AND `text` like ('%puodukas%'))
)
```

Iš akmens amžiaus prisiminimų

Kaip spręsti?

- **Sphinx**
- Apache Solr / Lucense
- Xapian

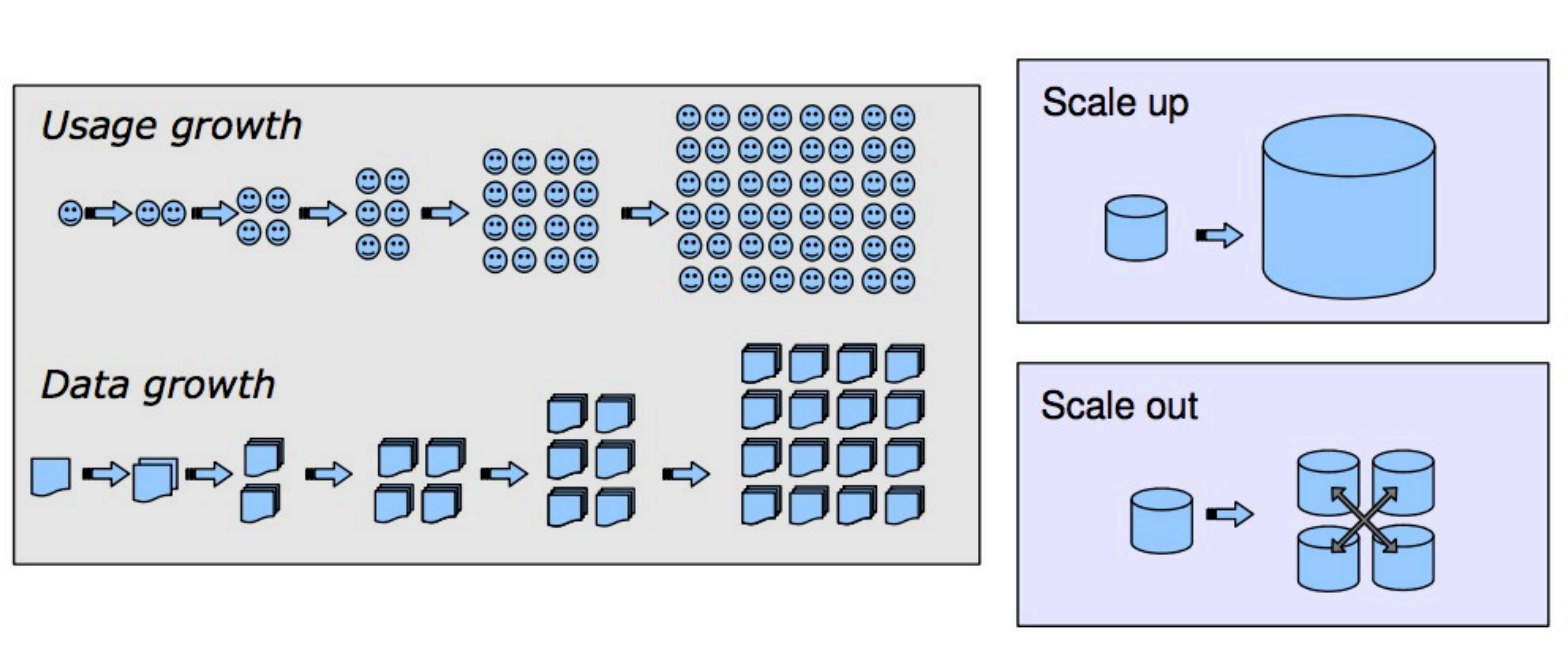
Kas tas Sphinx’as?

- Vystoma nuo 2001
- Atviro kodo / GPLv2
- C++
- Mysql protokolo palaikymas / SQL užklausos
- Paprasta integruoti
- Lengvas konfigūravimas
- Galima plėsti tiek horizontaliai tiek vertikaliai (daugiau serverių)

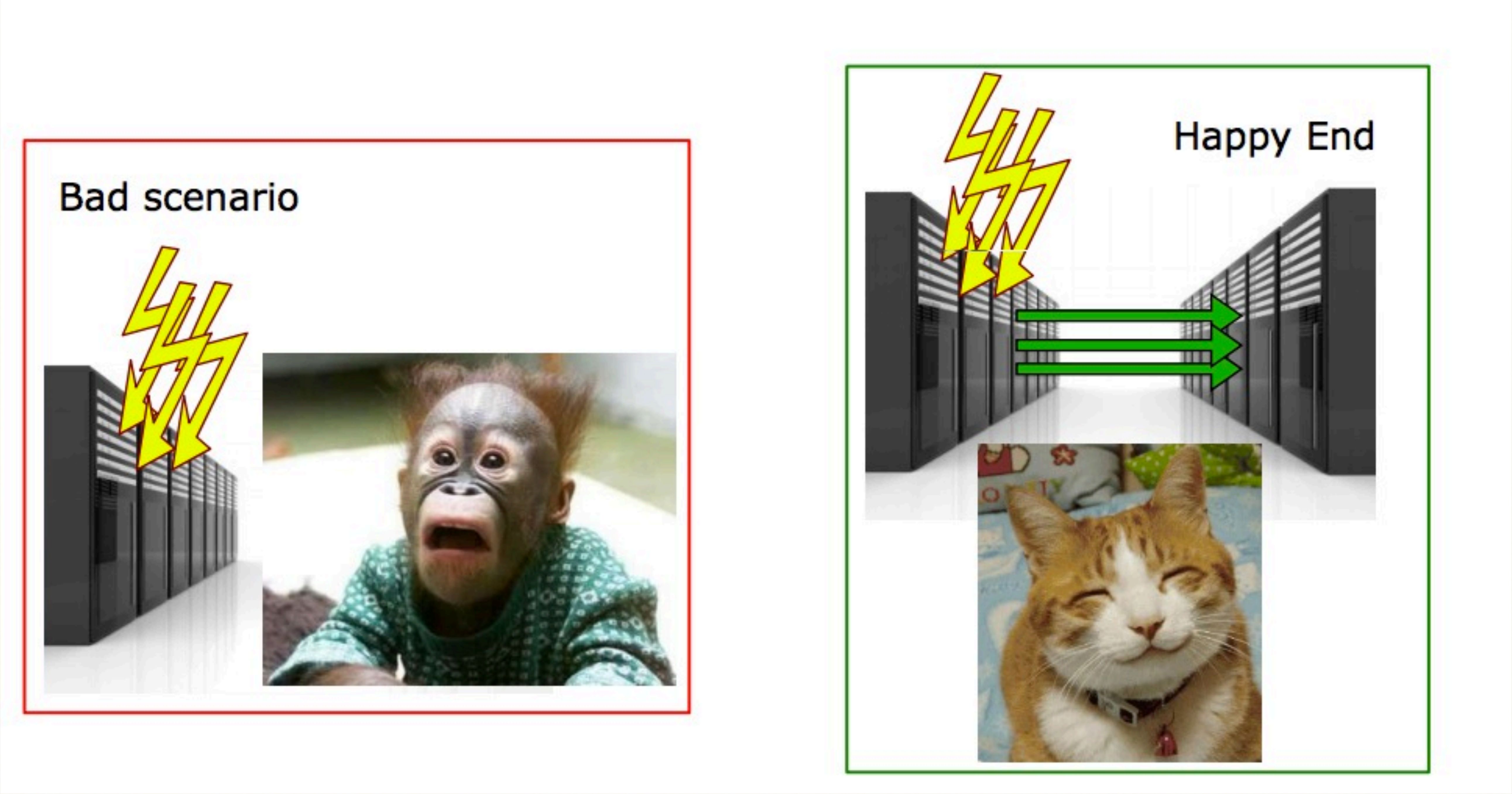
Šiek tiek faktų (iš sphinx saito):

- Duomenų indeksavimas: 10-15MB/s (teksto)
- Paieška: 1000000 dokumentų (1.2GB) - 500 užklausų / s
- Didžiausia man žinoma sistema turi daugiau nei 50TB index’a.

Scalability



High-availability



Instaliavimas

```
./configure --prefix=/home/zilionis/sphinx --enable-id64
./configure --without-mysql --with-pgsql --enable-id64
make
make install
```

Konfiguracija *sphinx.conf*

- Source - iš kur duomenys gaunami. Realiai tai, bet kokios validžios užklausos rezultatas
- Index - gali būti naudojamas daugiau nei vienas source.
- searchd - deamono konfigūracija

Konfigūracija: Minimalus pavyzdys

-
  ```
  source min {
   type = mysql
   sql_host = localhost
   sql_user = root
   sql_pass = slaptaszodis
   sql_db = test
   sql_query = select 1, 'cat' union select 2, 'dog'
  }
  ```

-
  ```
  index idx_min {
    path = idx
    source = min
  }
  ```

-
  ```
  searchd {
    listen = 9306:mysql41
    log = sphinx.log
    pid_file = sphinx.pid
  }
  ```

Indeksavimas

```
./indexer -c sphinx.conf --all
Sphinx 2.0.6-id64-release (r3473)
Copyright (c) 2001-2012, Andrew Aksyonoff
Copyright (c) 2008-2012, Sphinx Technologies Inc (http://sphinxsearch.com)

using config file 'sphinx.conf'...
indexing index 'idx_min'...
WARNING: Attribute count is 0: switching to none docinfo
collected 2 docs, 0.0 MB
sorted 0.0 Mhits, 100.0% done
total 2 docs, 6 bytes
total 0.021 sec, 284 bytes/sec, 94.69 docs/sec
total 2 reads, 0.000 sec, 0.0 kb/call avg, 0.0 msec/call avg
total 6 writes, 0.000 sec, 0.0 kb/call avg, 0.0 msec/call avg
```

- **./indexer -c sphinx.conf --all --rotate**

Paieška

```
» ./search -c sphinx.conf dog
Sphinx 2.0.6-id64-release (r3473)
Copyright (c) 2001-2012, Andrew Aksyonoff
Copyright (c) 2008-2012, Sphinx Technologies Inc (http://sphinxsearch.com)

using config file 'sphinx.conf'...
index 'idx_min': query 'dog ': returned 1 matches of 1 total in 0.010 sec

displaying matches:
1. document=2, weight=1643

words:
1. 'dog': 1 documents, 1 hits
```

Paieška

```
» ./search -c sphinx.conf "dog|cat"
Sphinx 2.0.6-id64-release (r3473)
Copyright (c) 2001-2012, Andrew Aksyonoff
Copyright (c) 2008-2012, Sphinx Technologies Inc (http://sphinxsearch.com)

using config file 'sphinx.conf'...
index 'idx_min': query 'dog|cat ': returned 2 matches of 2 total in 0.000 sec

displaying matches:
1. document=1, weight=1571
2. document=2, weight=1571

words:
1. 'dog': 1 documents, 1 hits
2. 'cat': 1 documents, 1 hits
```

Daemono paleidimas

```
» ./searchd -c sphinx.conf
Sphinx 2.0.6-id64-release (r3473)
Copyright (c) 2001-2012, Andrew Aksyonoff
Copyright (c) 2008-2012, Sphinx Technologies Inc (http://sphinxsearch.com)

using config file 'sphinx.conf'...
WARNING: compat_sphinxql_magics=1 is deprecated; please update your application and config
listening on all interfaces, port=9306
precaching index 'idx_min'
precached 1 indexes in 0.000 sec

» mysql -hlocalhost -P9306 --protocol=tcp
------------------------------------------------------------------------------------------------------------------------------------
mysql> select * from idx_min where match('cat');
+------+--------+
| id   | weight |
+------+--------+
|    1 |   1643 |
+------+--------+
1 row in set (0.00 sec)
```

Konfigūracijos būdai

- Single index
- Main + delta scheme
- Multiple indexes
- Multiple Sphinx instances
- Sphinx Search Cluster
- Real Time indeksai

Konfigūracija: Paveldejimas

```
source text1
{
        type            = mysql
        sql_host        = localhost
        sql_user        = b
        sql_pass        = u
        sql_db          = b
        sql_port        = 3306
        sql_query       = select id, body, published, lat, long, category from table
        sql_attr_timestamp      = published
        sql_attr_float  = lat
        sql_attr_float  = long
        sql_attr_uint   = category
}

source text2 : text1
{
        sql_query       = select id, user_name, inserted from table2
        sql_attr_timestamp      = inserted
        sql_attr_float  =
        sql_attr_uint   =
}
```

Konfigūracija: Generavimas

```
#!/usr/bin/php
<?php
$m = new mysqli('maindb', 'user', 'password', 'main');
$res = $m->query("select site_map.id, ip from site_map left join server on site_map.master_id = server.id");

while ($row=$res->fetch_assoc()) {
        $n = $row['id'];
        $host = $row['ip'];
        echo "
source chunk{$n} {
    type = mysql
    sql_host = {$host}
    sql_user = user
    sql_pass = pass
    sql_db = c{$n}
    sql_query_pre = SET NAMES utf8
    sql_query = select id, {$n} chunk_id, body from a{$n} where id>=\$start AND id<=\$end and crawled=0
    sql_query_range     = SELECT MIN(id),MAX(id) FROM a$n
    sql_range_step = 100000
}
";
}
```

Konfigūracija: Main source

```
source dbbl2_msg_000
{
   type        = mysql
   sql_host     = dbbl2-local
   sql_user = nnseek
   sql_pass =
   sql_db = nn2_msg000
   sql_query_pre = SET NAMES utf8
   sql_query = SELECT \
               m.id, m.group_id, m.language_id, g.def_lang_id as  grp_def_lang_id, unix_timestamp(m.ts) as ts, unix_timestamp(m.published)  as published,
m.subject, uncompress(m.message) as message,
         FROM \
               nn2_nnseek.grp g, msg000 m, nn2_nnseek.nnauthors a \
         WHERE \
               m.id>=$start and m.id<=$end AND g.active = 1 AND g.do_index = 1 AND g.hidden != 1 AND g.id=m.group_id AND m.deleted=0 AND m.author_id
a.id
   sql_query_range  = SELECT MIN(id),MAX(id) FROM msg000 WHERE id > 0
   sql_ranged_throttle = 175
   sql_range_step   = 50000

   sql_query_post_index = UPDATE nn2_nnseek.index SET last_indexed_msgid = $maxid, index_time = NOW() WHERE source_id = '0'
   sql_attr_uint = group_id
   sql_attr_uint = language_id
   sql_attr_uint = grp_def_lang_id.
   .....
}
```

Konfigūracija: Delta source

```
source dbbl2_delta_msg_000
{
   .....
   sql_query_pre = SET NAMES utf8
   sql_query = SELECT \
               m.id, m.group_id, m.language_id, g.def_lang_id as  grp_def_lang_id, unix_timestamp(m.ts) as ts, unix_timestamp(m.published)  as published,
m.subject, uncompress(m.message) as message,
         FROM \
               nn2_nnseek.grp g, msg000 m, nn2_nnseek.nnauthors a \
         WHERE \
               m.id>=$start and m.id<=$end AND g.active = 1 AND g.do_index = 1 AND g.hidden != 1 AND g.id=m.group_id AND m.deleted=0 AND m.author_i
a.id
   sql_query_range  = SELECT  last_indexed_msgid, max(nn2_msg000.msg000.id) FROM nn2_nnseek.index, nn2_msg000.msg000 where source_id=0
   sql_ranged_throttle> = 175
   sql_range_step  = 50000

   sql_query_post_index = UPDATE nn2_nnseek.index SET last_indexed_msgid_small = $maxid, index_time_small = NOW() WHERE source_id = '0'
   .....
}
```

Indeksas

```
index dbbl2_delta_msg_part3
{
   path              = /mnt/data/nnseek.sphinx/data/dbbl2_delta_msg_part3
   morphology              = stem_enru
   stopwords             = /mnt/data/nnseek.sphinx/stopwords.txt
   charset_type          = utf-8
   html_strip           = 1

        source              = dbbl2_delta_msg_064
        source              = dbbl2_delta_msg_065
        source              = dbbl2_delta_msg_066
        source              = dbbl2_delta_msg_067
        source              = dbbl2_delta_msg_068
 ....
}
```

Indeksas

```
index nn2_nnseek_related
{
type = distributed
local = rt_local
agent = ddbal1:3314:rt_local
agent = ddbal2:3314:rt_local
....
agent_connect_timeout   = 300
agent_query_timeout     = 300000
}
```

Real time indexas

```
index rt
{
   type = rt
   path = /usr/local/sphinx/data/rt
   rt_field = title
   rt_field = content
   rt_attr_uint = gid
}
```

Real time indexas

```
mysql> INSERT INTO rt VALUES ( 1, 'first record', 'test one', 123 );
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO rt VALUES ( 2, 'second record', 'test two', 234 );
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM rt;
+------+--------+------+
| id   | weight | gid  |
+------+--------+------+
|    1 |      1 |  123 |
|    2 |      1 |  234 |
+------+--------+------+
2 rows in set (0.02 sec)

mysql> SELECT * FROM rt WHERE MATCH('test');
+------+--------+------+
| id   | weight | gid  |
+------+--------+------+
|    1 |   1643 |  123 |
|    2 |   1643 |  234 |
+------+--------+------+
2 rows in set (0.01 sec)

mysql> SELECT * FROM rt WHERE MATCH('@title test');
Empty set (0.00 sec)
```

```
DELETE FROM rt WHERE id=2;
REPLACE INTO rt VALUES ( 1, 'first record on steroids',
'test one', 123 );

Select count(*) from rt
```

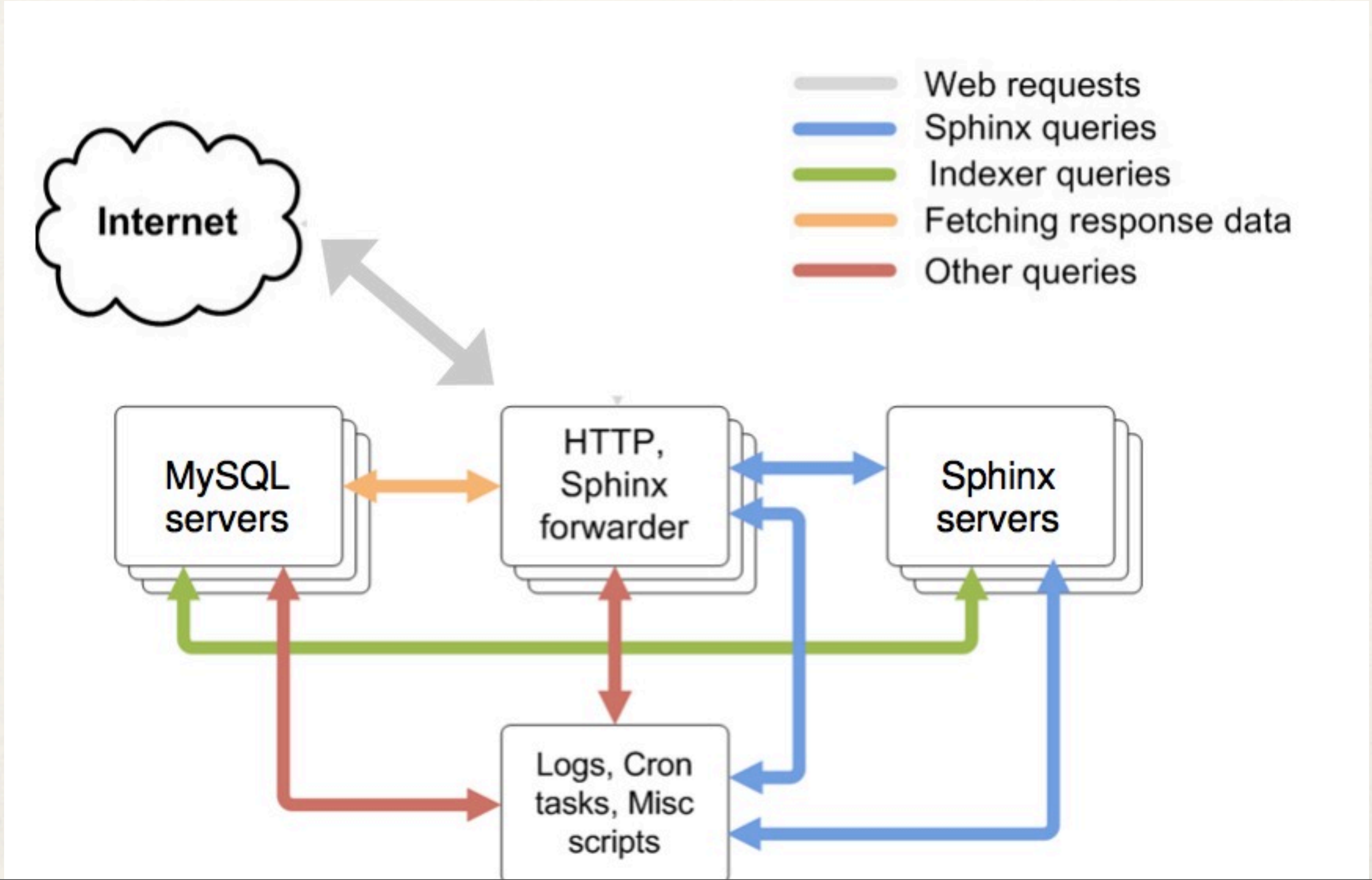
```
mysql> select * from idx_min where match('cat|dog');
+------+--------+
| id   | weight |
+------+--------+
|    1 |   1571 |
|    2 |   1571 |
+------+--------+
2 rows in set (0.00 sec)

mysql> show meta;
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| total         | 2     |
| total_found   | 2     |
| time          | 0.000 |
| keyword[0]    | cat   |
| docs[0]       | 1     |
| hits[0]       | 1     |
| keyword[1]    | dog   |
| docs[1]       | 1     |
| hits[1]       | 1     |
+---------------+-------+
9 rows in set (0.00 sec)
```

Indeksas: Plain Text + Real Time

```
index distributed
{
 type = distributed
 local = plain_main_index
 local = real_time_increment_index
}
```

Architektūros pavyzdys



Kaip veikia aplikacijos’e?

- Vykdoma paieška Sphinx’o indeks’e
- Gaunami reikalingi ID (atitinkantys užklausą) rezultatai
- Šiuos ID, pasiunčiam į Mysql ir gaunam mums reikalingus objektus

Kaip veikia aplikacijos’e?

- `mysql> select * from idx_min where match ('text | new') ;`

```
+------+--------+
| id | weight |
+------+--------+
| 10 | 1588 |
| 8 | 1568 |
| 2 | 1520 |
| 4 | 1520 |
| 14 | 1520 |
+------+--------+
5 rows in set (0.00 sec)
```

- `mysql> select * from some_table where id in (10,8,2,4,14);`

```
+----+-----------+
| id | some_text |
+----+-----------+
| 2 | test text |
| 4 | text test |
| 8 | new row |
| 10 | new text |
| 14 | old text |
+----+-----------+
```

Kaip veikia aplikacijos’e?

```
     mysql> select * from some_table where id in (10,8,2,4,14) ORDER BY FIELD(id, 10,8,2,4,14);
+----+-----------+
| id | some_text |
+----+-----------+
| 10 | new text |
| 8 | new row |
| 2 | test text |
| 4 | text test |
| 14 | old text |
+----+-----------+
```

BuildExcerpts

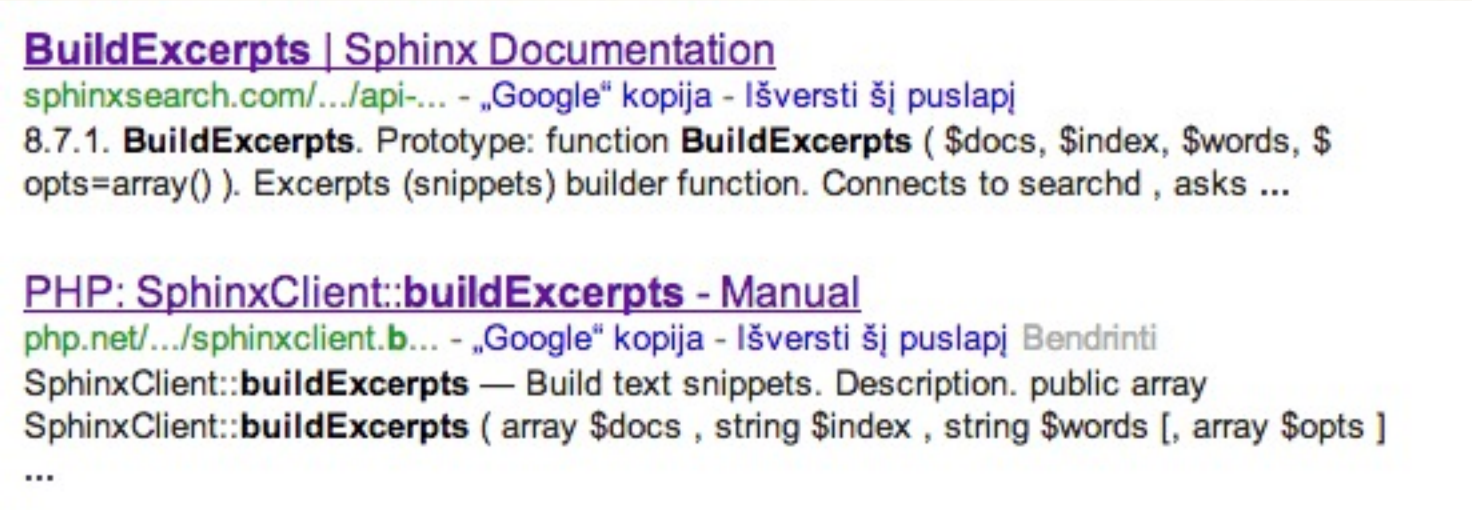
```
function buildExcerptFile($documents, $options = array())
{
        foreach($documents as $doc){
            $file = "/space/".'snip_'.md5($doc).'_'.time();
            file_put_contents($file, $doc);
            $files[] = $file;
        }

        $client = new SphinxClient();
        $client->setServer('localhost', 9312);

        $res = $client->BuildExcerpts( $files, 'index', $keywords,
                array(
                    'around'=>10,
                    'limit' => 300,
                    'load_files' => 1
                    )
                );

        foreach($files as $file){
            unlink($file);
        }

        return $res;
}
```



BuildExcerpts | Sphinx Documentation

sphinxsearch.com/.../api-... - „Google“ kopija - Išversti šį puslapį

8.7.1. **BuildExcerpts**. Prototype: function **BuildExcerpts** ($docs, $index, $words, $ opts=array()). Excerpts (snippets) builder function. Connects to searchd , asks ...

PHP: SphinxClient::**buildExcerpts** - Manual

php.net/.../sphinxclient.**b**... - „Google“ kopija - Išversti šį puslapį Bendrinti

SphinxClient::**buildExcerpts** — Build text snippets. Description. public array SphinxClient::**buildExcerpts** (array $docs , string $index , string $words [, array $opts] ...

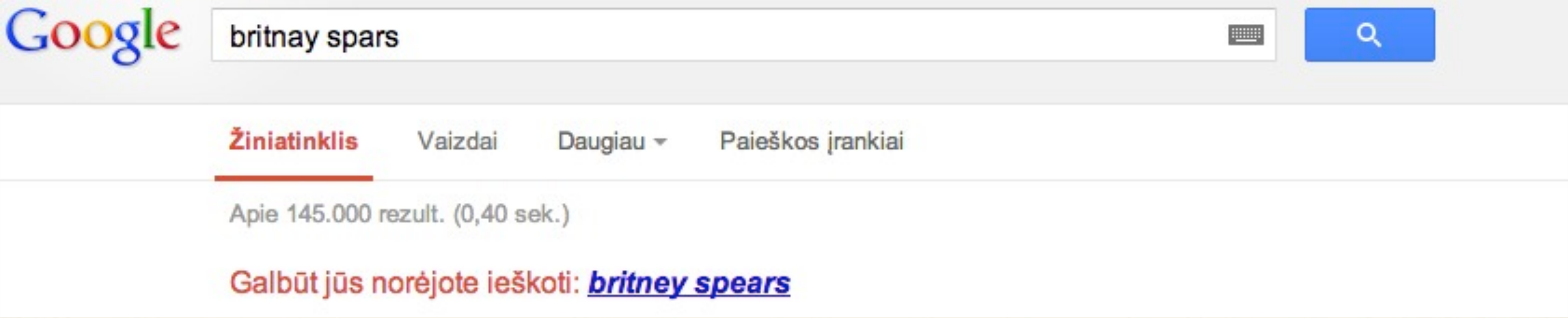
Perfomanco patarimai

- Indeksuok tik tai, ką naudosi paieškai
- Jei duomenų bazėje kompresuoti duomenis, naudok nustatyma unpack_mysqlcompress, tai leis atlikti Sphinxo pusėje - sutaupysi CPU / Tinklo resursus
- Naudoti kuo mažesnius indeksus
- Kai reikalinga - skaidyk didelius indeksus į mažesnius
- Naudok ranged queries, tai leis lengviau Mysql kvėpuoti
- Limituok sphinx'o atributus

Mintys pabaigai

- 3-čių šalių paieškos variklius yra gan sudėtinga prižiūrėti. Pagrindinė problema - duomenų šviežumas. Žinoma prisideda papildomi rūpesčiai, kaip papildomo softo instaliavimas, priežiūra
- Duomenų bazėje esantys FULLTEXT yra gėris, net ir jei šis sprendimas nėra greičiausias
- Skirtingi paieškos implementavimai gali pateikti skirtingus rezultatus, tad nereikia toleruoti tik vieno sprendimo. Geriausiai pasirinkti tai - kas tinka jūsų projektui
- Bet kokiu atveju, bet koks paieškos sprendimas yra daug geresnis nei "akmens amžiaus" LIKE :)

Namų darbų norit?



britnay spars

Žiniatinklis Vaizdai Daugiau Paieškos įrankiai

Apie 145.000 rezult. (0,40 sek.)

Galbūt jūs norėjote ieškoti: ***britney spears***

- Parsiųskit sphinx source failus
- misc/suggest/ - atsakymas :)

Nuorodėlės

- http://sphinxsearch.com/
- http://www.ivinco.com/blog/
- Google :D