Об опыте использования SphinxSearch в информационной системе

Дмитрий Воронин
АО НПП Кибертехника
18 мая 2018

Задача

- Реализовать поиск (атрибутный, полнотекстовый, смешанный) документов формата html (doc, docx, xml)

Требования

- PostgreSQL 9.4.5
- Кроссплатформенность (Linux, Windows)
- Время на отклик — 6 мс
- No Java (sic!)

Атрибутный поиск

- Найти документ, удовлетворяющего метрикам:
  - дата создания
  - тип документа
  - автор
  - ключевые слова
  - и т. д.

Полнотекстовый поиск

- Требуется реализовать по телу документа и названию
- Должны поддерживаться:
  - конструкции И, ИЛИ, НЕ
  - фразовый поиск
  - поиск на двух языках (русский и английский)

Пример документа

```
возможностями</span></font></font><font face="Calibri, serif"><font size="2" style="font-size: 10pt"><b>
</b></font></font><font face="Calibri, serif"><font size="2" style="font-size: 10pt">подразумевается
наибольший объем выпуска продукции,
который достигается при полном
использовании ресурсов, т.к. ресурсы
ограничены, то общество вынуждено делать
технический </font></font><font face="Calibri, serif"><font size="2" style="font-size: 10pt"><b>выбор</b></font></font><font
face="Calibri, serif"><font size="2" style="font-size: 10pt">,
решая </font></font><font face="Calibri, serif"><font size="2" style="font-size: 10pt"><b>какие
из потребностей следует удовлетворить,
а какие нет</b></font></font><font face="Calibri, serif"><font size="2" style="font-size: 10pt">.</font></font></font></font></
p>
<p style="margin-bottom: 0cm; line-height: 100%"><font face="Times New Roman, serif"><font size="3" style="font-size: 12pt"><
font face="Calibri, serif"><font size="2" style="font-size: 10pt">Экономическая
теория констатирует, что ограниченность
ресурсов определяет альтернативность
их использования и общество из всех
вариантов выбирает наиболее оптимальный.</font></font></font></font></p>
<p style="margin-bottom: 0cm; line-height: 100%"><font size="2" style="font-size: 10pt"><b>Альтернативная
стоимость</b></font><font size="2" style="font-size: 10pt">
- стоимость </font><font size="2" style="font-size: 10pt">товара</font><font size="2" style="font-size: 10pt">
или услуги, измеряемая через </font><font size="2" style="font-size: 10pt">цену</font><font size="2" style="font-size: 10pt">
др.товара или услуги, от приобретения
которых мы вынуждены отказаться, причём
имеется в виду наилучший товар или
услуга, от которой мы должны отказаться,
отдавая предпочтение данному варианту
использования наших </font><font size="2" style="font-size: 10pt">денег</font><font size="2" style="font-size: 10pt">,
```

Почему не FTS PostgreSQL?

- tsvector не поддерживает слишком большие строки
- Для индексирования данных требуется предварительная обработка
- Не поддерживается поиск сразу по нескольким языкам
- Непредсказуемое время запроса

Почему SphinxSearch?

- OpenSource
- Самый функциональный и известный, написанный на C/C+
- Хорошая скорость построения индексов
- SQL язык запросов
- Нативная поддержка PostgreSQL
- Доступ по протоколу MySQL
- На момент работы над проектом — взяли версию 2.2.11

Смешанный поиск

- Выполнить полнотекстовый поиск по телу документа и наложить условия атрибутного
- Использовали драйвер pymysql для доступа в Sphinx
- Psycopg2 для доступа в PostgreSQL
- Хочется использовать **один драйвер** для получения результатов смешанного поиска

Из PostgreSQL в SphinxSearch

- pg_sphinx:
  - sphinx_select() возвращает (id, weight)
  - нет возможности возвратить что-то свое
  - нет возможности написать произвольный запрос
- That’s all, folks!

sphinxlink

- Идея взята из расширения dblink
- Подключения сохраняются в статическом HTAB бекэнда PostgreSQL
- Доступ к SphinxSearch реализован с помощью C-API MySQL (MariaDB)
- Автоподключение к SphinxSearch при разрыве соединения с ним

sphinxlink (2)

```
sphinx_connect(conname TEXT,
               host TEXT DEFAULT '127.0.0.1',
               port INTEGER DEFAULT 9306)

sphinx_query(conname TEXT, query TEXT)

sphinx_meta(conname TEXT, OUT varname TEXT, OUT value TEXT)

sphinx_connections(OUT conname TEXT,
                   OUT host TEXT,
                   OUT port INTEGER)

sphinx_disconnect(conname TEXT)
```

Примеры

```
SELECT * FROM sphinx_connect('myconn');
  sphinx_connect
  ----------------
   OK
 (1 строка)

SELECT * FROM sphinx_connections();
  conname |   host        | port
  ------------+-------------+------
  myconn   | 127.0.0.1 | 9306
(1 строка)
```

Примеры (2)

```
SELECT * FROM sphinx_query('myconn',
                           'SELECT weight(), * FROM myindex
                                   WHERE MATCH(''красота'')')
AS (weight integer, docid integer, title text);
 weight | docid | title
--------+-------+-------
   1680 |     3 | Дома
(1 строка)
```

Примеры (3)

```
SELECT * FROM sphinx_meta('myconn');
   varname    | value
--------------+--------
 total        | 1
 total_found  | 1
 time         | 0.000
 keyword[0]   | красот
 docs[0]      | 1
 hits[0]      | 1
(6 строк)
```

Примеры (4)

```
SELECT * FROM sphinx_disconnect('myconn');
 sphinx_disconnect
-------------------
 OK
(1 строка)
```

sphinxlink: TODO

- Хранение параметров подключений к SphinxSearch в специальной таблице
- Поддержка RT индексов
- Параметризованные запросы?
- Ваши идеи?

SphinxSearch: актуализация индексов

- Внешнего API нет
- Рецептов нет :(
- Удаленный запуск indexer
  - Требуется подготовить конфигурацию
  - Хочется как-то убедиться в результате действия

sphinx_agent

- Демон, реализован на Python 2.7
- Управляет SphinxSearch
- Опрашивает таблицу на предмет изменений (периодически)
- Содержит генератор конфигурации
- Выполняет рутинную работу в нескольких потоках

Хотелки для SphinxSearch

- Управление составом индексов через SQL или же через HTTP API
- Build-in репликации индексов
- Средство резервирования и восстановления индексов

Спасибо за внимание!

Вопросы?

http://github.com/dimv36/sphinxlink

https://habr.com/post/347700/