Об эффективности поиска данных в веб-приложениях

А.Н. Земцов[1], *Зунг Хань Чан*[2]

[1]*Волгоградский государственный технический университет*
[2]*Национальный экономический университет*

Аннотация: Рассматривается важная составляющая систем электронного документооборота и обучения – механизм полнотекстового поиска, позволяющий реализовывать удобные средства поиска интересующей информации по содержимому электронных документов. Приводятся оценки эффективности полнотекстового поиска в базах данных.

Ключевые слова: Базы данных, полнотекстовый поиск, система управления базами данных, MySQL, PostgreSQL, Oracle.

Организация полнотекстового поиска, расширение возможностей индексирования, а также оценка эффективности их применения в различных СУБД, являются важными этапами создания современных систем, работающих с электронными документами. Под полнотекстовым поиском понимается поиск по содержимому документов базы данных системы электронного документооборота [1], а также совокупность концептуальных подходов и методов к оптимизации этого процесса [2].

Использование механизма полнотекстового поиска в современных системах электронного документооборота, в том числе с применением онтологий [3], создает для пользователей более широкие возможности поиска информации в базе данных, по сравнению с традиционным поиском по названию документа и ключевым словам [2]. Такой подход позволяет обеспечить пользователю возможность получения лучшей связности материала, а также эффективную навигацию по нему [4, 5].

Полнотекстовый поиск является важной составляющей различных информационных систем: систем электронного документооборота и обучения [6,7], веб-систем широкого спектра, систем биометрического контроля доступа [8, 9] и т.д.

Оценка эффективности выполнения поисковых запросов в СУБД MySQL, PostgreSQL и Oracle производилась с помощью тестовой таблицы, содержащей только ключевое поле id и текстовое поле content, по которому выполнялся полнотекстовый поисковый запрос с ранжированием результатов.

Ниже показан пример создания тестовой таблицы в СУБД MySQL:

```
create table 'test1000' ('id' int(11) not null auto_increment, 'content' text character set utf8 collate utf8_bin, primary key ('id'), fulltext key 'content' ('content')) default charset=utf8
```

Создадим соответствующую таблицу в СУБД PostgreSQL:

```
create table test1000 (id serial not null, content text, primary key (id))
```

Создание тестовой таблицы в СУБД Oracle запишется в виде:

```
create table test1000 (id number primary key, content varchar2(200))
```

Дополнительно создадим индекс для тестовой таблицы:

```
create index test_content_idx on test1000 (content) index type is ctxsys.context;
```

Выполнение запроса в СУБД PostgreSQL может производиться как с использованием, так и без использования индекса [10]. В связи с этим, представляет интерес дополнительно рассмотреть вариант организации таблицы с полем fts типа tsvector, для которого создадим GIN-индекс:

```
create index fts_idx on test1000 using gin(fts);
```

Тип данных tsvector используется как хранилище для лексем, помимо которых может хранить сведения о месте лексемы в документе и ее весе, который может использоваться для ранжирования результатов [11].

Поисковые запросы для тестовых баз данных будут выглядеть соответствующим образом. В СУБД MySQL запрос запишется в виде:

```
select id, content, match (content) against ('Россия') as score from test1000 where match (content) against ('Россия') order by score desc
```

В СУБД PostgreSQL без использования индекса:

select id, content, ts_rank (to_tsvector(content), q) from test1000, to_tsquery('Россия') q where to_tsvector(content) @@ q order by ts_rank desc

В СУБД PostgreSQL с использованием индекса:

select id, content, rank_cd (fts, q) from test1000, to_tsquery('Россия') q where fts @@ q order by rank_cd desc

Поисковый запрос в СУБД Oracle:

select id, content, score (1) from test1000 where contains (content, 'Россия', 1) > 0 order by score (1) desc

Для оценки эффективности выполнения поисковых запросов рассмотрим 9 вариантов тестовой таблицы: с количеством записей равным $N = \{10^2, 10^3, 10^4\}$, каждая из которых может содержать k ключевых слов в поле content, где $k = \{1, 5, 10\}$.

Поскольку СУБД MySQL и СУБД Oracle для реализации поискового запроса по текстовому полю требуют наличия индекса, то будем сравнивать их с СУБД PostgreSQL с учетом выполнения требования индексации данных.

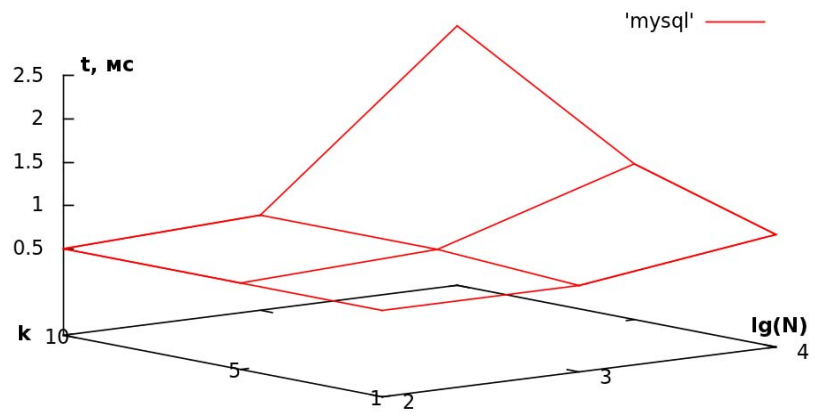

Рис. 1. –Производительность выполнения запроса в СУБД MySQL.

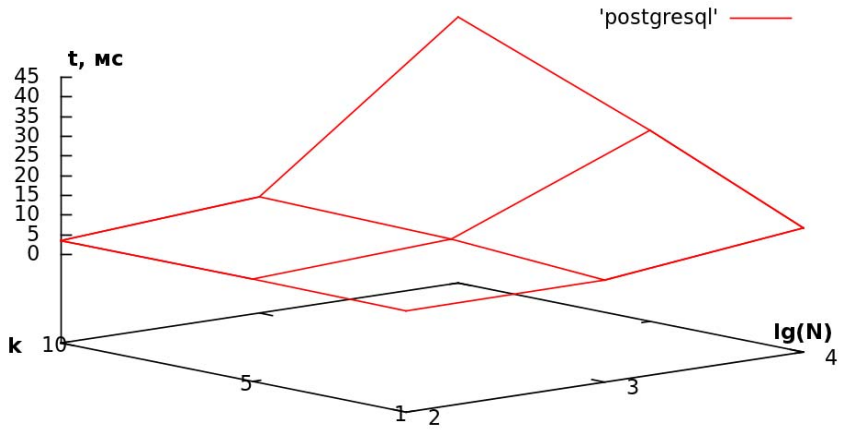

Рис. 2. –Производительность выполнения запроса в СУБД PostgreSQL.

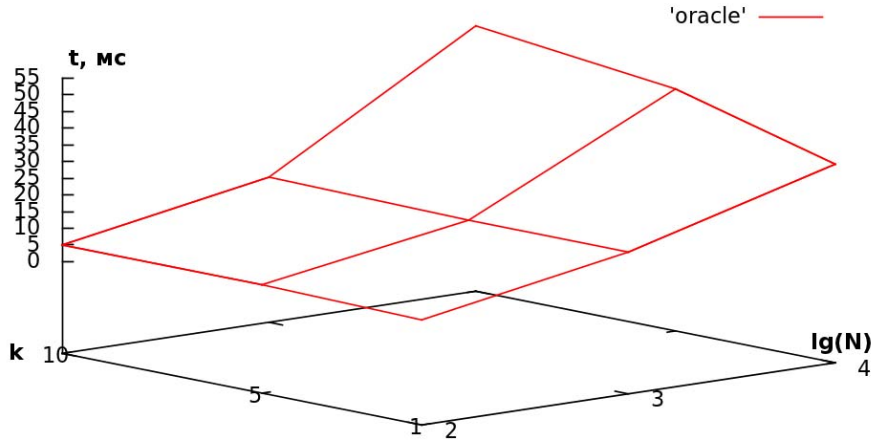

Рис. 3. –Производительность выполнения запроса в СУБД Oracle.

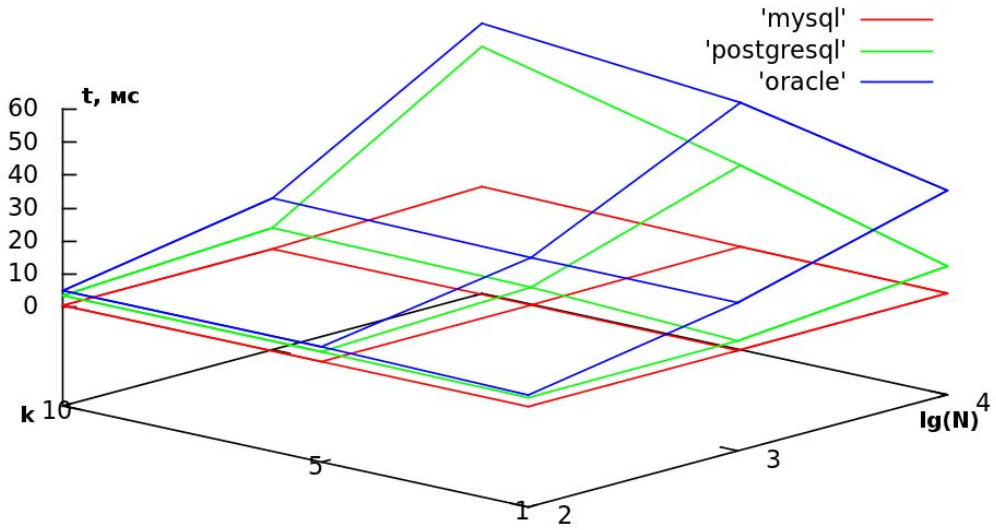

Рис. 4. –Сравнение производительности различных СУБД.

Полученные в результате исследования данные для наглядности представим в виде семейства кривых, образующих поверхность. Осями абсцисс и ординат служат количество ключевых слов k и количество записей $lg(N)$, соответственно.

Для отображения зависимостей, показанных на рис.1-3, используется собственный масштаб по оси аппликат, поэтому для сравнения эффективности поиска различных СУБД между собой приведем результаты в одном масштабе на рис. 4.

В данном случае логарифмическая шкала для N удобнее для анализа результатов и дает более показательный результат для зависимости времени выполнения поискового запроса t от количества записей N. Ось аппликат соответствует времени t в миллисекундах, затраченному на выполнение поискового запроса.

При малом количестве записей рассматриваемые СУБД характеризуются схожими временными затратами при выполнении поисковых запросов. Необходимо отметить, что время выполнения поискового запроса не зависит от длины полей с ключевыми словами k, но с увеличением количества записей N зависимость времени выполнения поискового запроса t от количества ключевых слов k становится более выраженной. Зависимость времени выполнения поискового запроса t от количества записей N при любом количестве ключевых слов k имеет логарифмический вид.

СУБД MySQL имеет более высокую эффективность поиска на представленных интервалах k и N, т.к. реализует упрощенную схему

полнотекстового поиска и ранжирования. Кроме того, СУБД MySQL в рассмотренных случаях показала лучшие результаты на малых объемах данных.

СУБД PostgreSQL с индексированием и СУБД Oracle демонстрируют примерно одинаковую производительность. Необходимо отметить, что СУБД Oracle незначительно уступает СУБД PostgreSQL, но имеет менее выраженную зависимость t от k с увеличением количества записей N, что скажется в пользу СУБД Oracle при количестве записей $N \geq 3 \cdot 10^4$.

Литература

1. Кацупеев А.А., Щербакова Е.А., Воробьев С.П., Литвяк Р.К. Модификация математической модели выбора оптимальной стратегии защиты распределённых систем// Инженерный вестник Дона, 2017, №1. URL:ivdon.ru/ru/magazine/archive/n1y2017/4078.
2. Земцов А.Н., Болгов Н.В., Божко С.Н. Многокритериальный выбор оптимальной системы управления базы данных с помощью метода анализа иерархий// Инженерный вестник Дона, 2014, № 2.URL: ivdon.ru/ru/magazine/archive/n2y2014/2360.
3. Евдошенко О.И., Кравец А.Г., Петрова И.Ю. Разработка онтологии и базы данных для эффективного поиска научно-технической информации // Прикладная информатика, 2015. Т.10. № 5. С. 85-92.
4. Лавриченко О.В. Разработка логико-концептуальной модели при принятии решений в теории экономики активного коннекта // Инженерный вестник Дона, 2015, № 1-2. URL: ivdon.ru/ru/magazine/archive/n1p2y2015/2834.
5. Седов В.А., Седова Н.А. Самооценка системы менеджмента качества с использованием теории нечетких множеств // Программные системы и вычислительные методы, 2014. № 4. С. 456-463.

6. Шапошников, Д.Е. Применение принципа гарантированного результата для учёта качественной информации о предпочтениях при комплексной оценке качества функционирования телекоммуникационных сетей // Инженерный вестник Дона, 2014, № 4-1. URL: ivdon.ru/ru/magazine/archive/N4y2014/2574.
7. Кацупеев А.А., Щербакова Е.А., Воробьев С.П. Постановка и формализация задачи формирования информационной защиты распределённых систем// Инженерный вестник Дона, 2015, № 1-2. URL: ivdon.ru/ru/magazine/archive/n1p2y2015/2868.
8. Zemtsov A.N. Robust audio stream protection method based on higher bits embedding // Nauka i studia. Przemysl (Poland), 2015. NR3(134). pp. 37-43.
9. Земцов А.Н. Методы цифровой стеганографии для защиты авторских прав. LAP Academic Publishing, 2012. 148 с.
10. Vitolo C. Web technologies for environmental Big Data // Environmental Modelling and Software, 2015. Vol. 63, pp. 185-198.
11. Rodríguez-García M.A. Ontology-based annotation and retrieval of services in the cloud // Knowledge-Based Systems, 2014. Vol.56. pp. 15-25.

References

1. Kacupeev A.A., Shherbakova E.A., Vorob'ev S.P., Litvjak R.K. Inženernyj vestnik Dona (RUS), 2017, № 1. URL: ivdon.ru/ru/magazine/archive/n1y2017/4078.
2. Zemtsov A.N., Bolgov N.V., Bozhko S.N. Inženernyj vestnik Dona (RUS), 2014. T. 29. № 2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2360.
3. Evdoshenko O.I., Kravec A.G., Petrova I.Ju. Prikladnaja informatika, 2015. T.10. № 5. pp. 85-92.
4. Lavrichenko O.V. Inženernyj vestnik Dona (RUS), 2015, № 1-2. URL: ivdon.ru/ru/magazine/archive/n1p2y2015/2834.

5. Sedov V.A., Sedova N.A. Programmnye sistemy I vychislitel'nye metody, 2014. № 4. pp. 456-463.
6. Shaposhnikov, D.E. Inženernyj vestnik Dona (RUS), 2014, № 4-1. URL: ivdon.ru/ru/magazine/archive/N4y2014/2574.
7. Kacupeev A.A., Shherbakova E.A., Vorob'ev S.P. Inženernyj vestnik Dona (RUS), 2014, № 4-1. URL:ivdon.ru/ru/magazine/archive/n1p2y2015/2868.
8. Zemtsov A.N. Nauka I studia. Przemysl (Poland), 2015. NR3 (134). pp. 37-43.
9. Zemtsov A.N. Metody cifrovoj steganografii dlja zashhity avtorskih prav [Methods of digital steganography for copyright protection]. LAP Academic Publishing, 2012. 148 p.
10. Vitolo C. Environmental Modelling and Software, 2015. Vol.63, pp.185-198.
11. Rodríguez-García M.A. Knowledge-Based Systems, 2014. Vol.56. pp.15-25.