



HighLoad

Производительность GIN и GiST индексов
в PostgreSQL

Федор Сигаев (PostgreSQL Team)

Индексы в базах данных

Индексы – главный метод ускорения поиска

- B-tree – для скалярных данных
- R-tree – для двумерных данных
- Bitmap – для данных с малой мощностью множества (cardinality)



Специализированные типы данных и задачи

- 3-мерные и более координаты
- Сферические координаты
- KNN (Nearest Neighbour Search)
- Массивы
- Поиск похожих слов
- Полнотекстовый поиск



Специндексы

B-tree

- Одна интерфейсная функция (compare)
- Операции: $<$, $<=$, $=$, $>=$, $>$
- GiST (Generalized Search Tree)
 - 7 интерфейсных функций (penalty, union, picksplit, consistent, same compress/decompress)
 - Операции: зависит от данных
- GIN (Generalize Inverted Index)
 - 4 интерфейсные функции (extractValue, extractQuery, compare, consistent)
 - Операции: зависит от данных

Полнотекстовый поиск

contrib/tsearch2 (8.0-8.2), встроенный в 8.3

Тип `tsvector` – представление документа, удобное для поиска, с нормализованными словами:

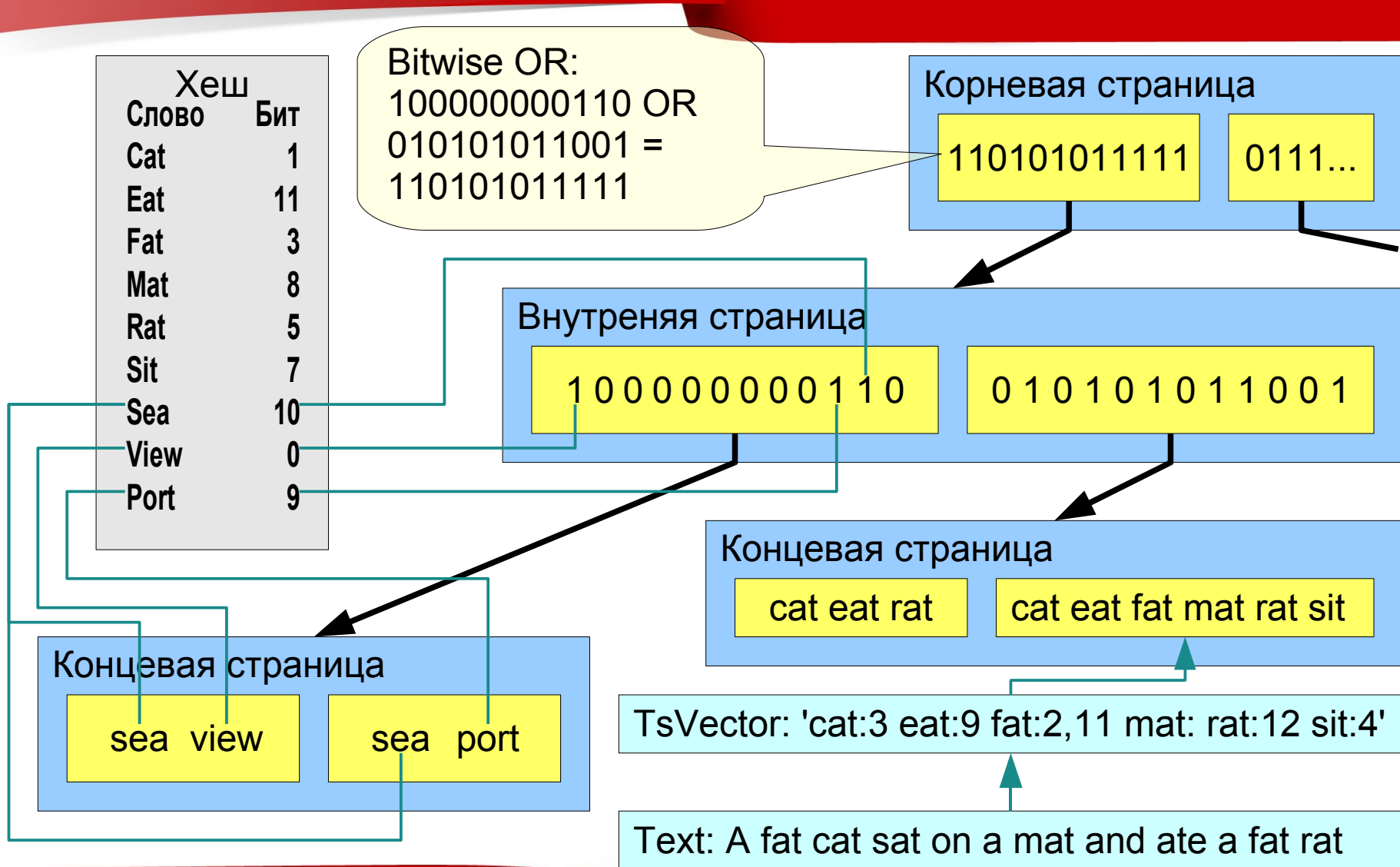
```
# select to_tsvector('A fat cat sat on a mat and ate a fat rat');
           to_tsvector
-----
'ate':9 'cat':3 'fat':2,11 'mat':7 'rat':12 'sat':4
```

`Tsquery` – полнотекстовый запрос

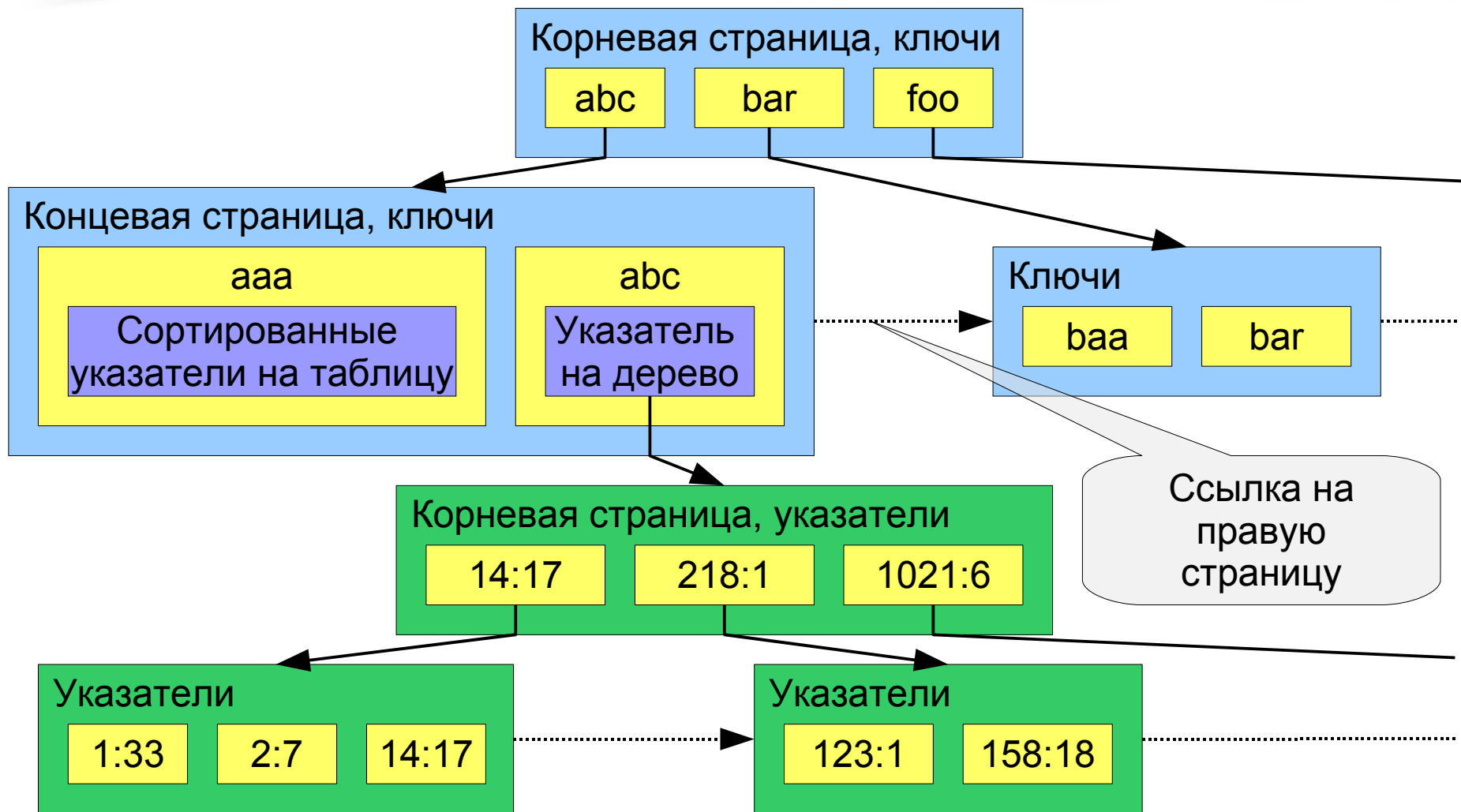
```
# select plainto_tsquery('fat cat');
 plainto_tsquery
-----
'fat' & 'cat'
```



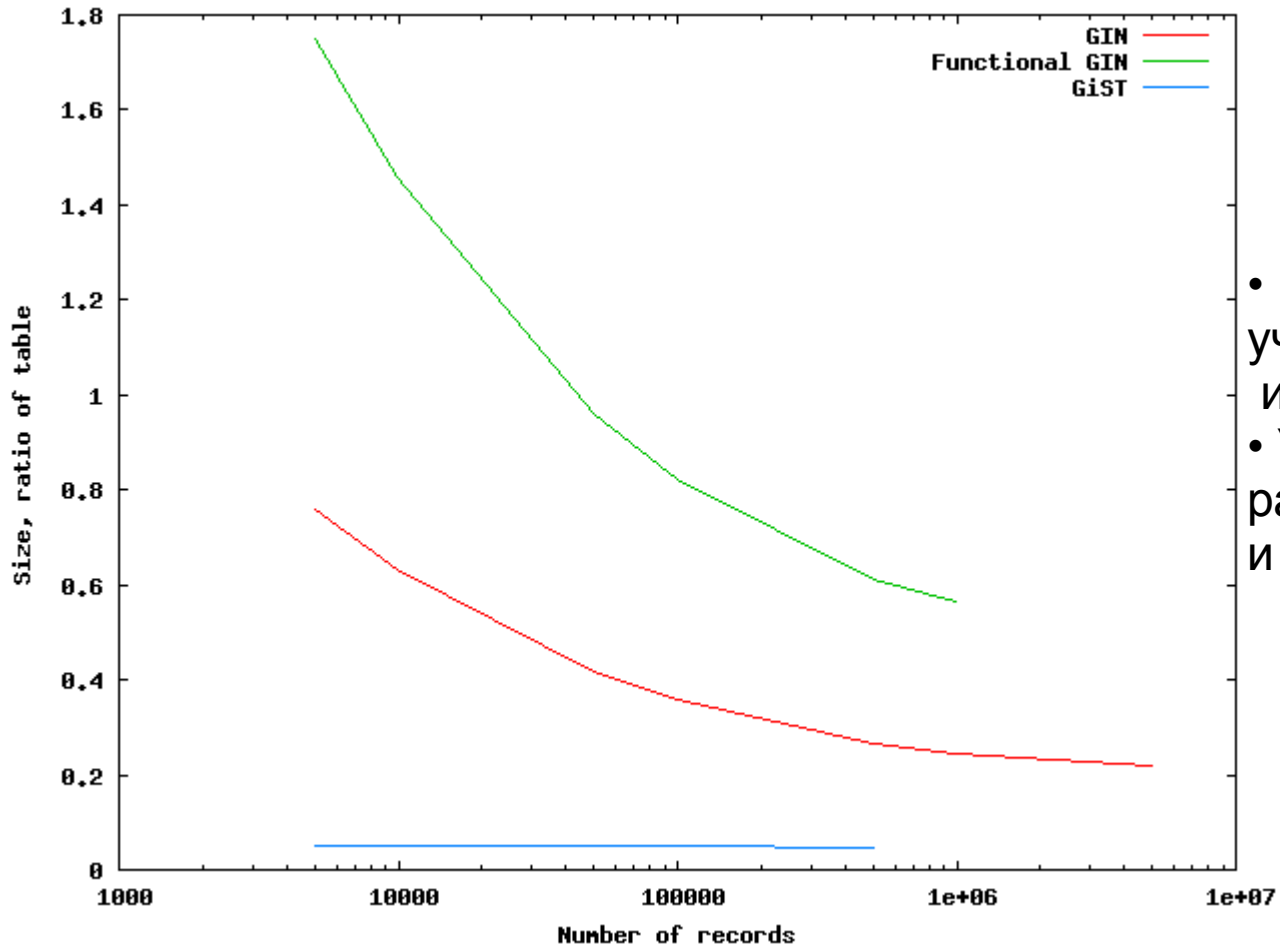
GiST для полнотекстового поиска



GIN для полнотекстового поиска

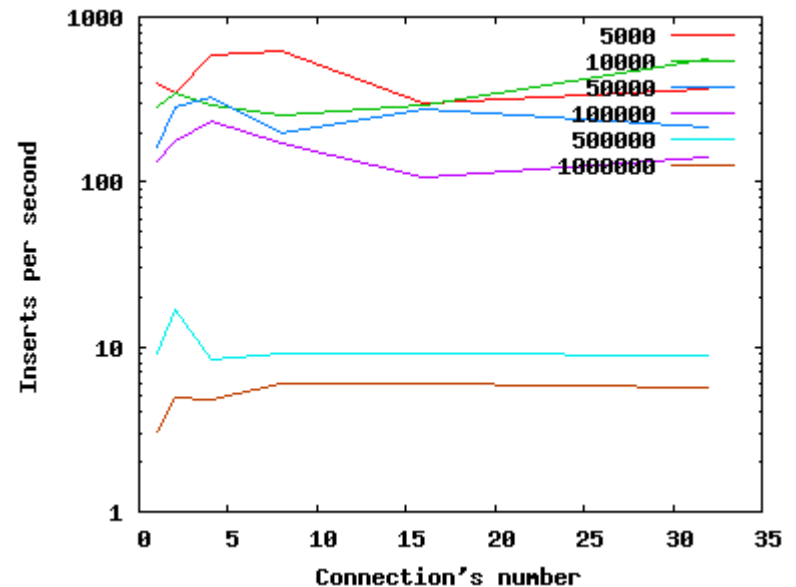
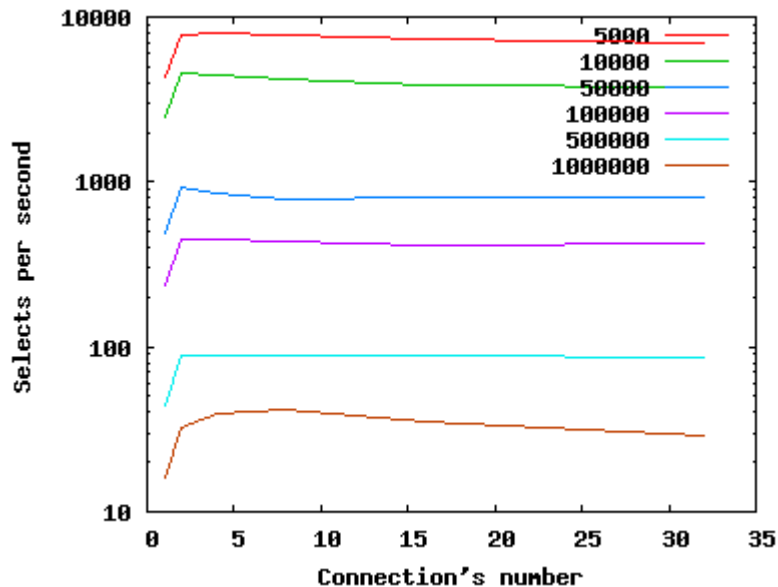


Размер индексов



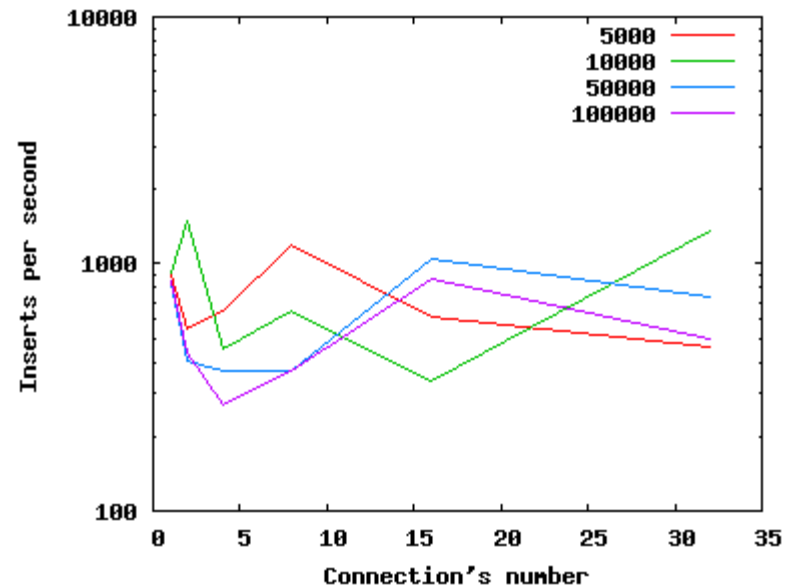
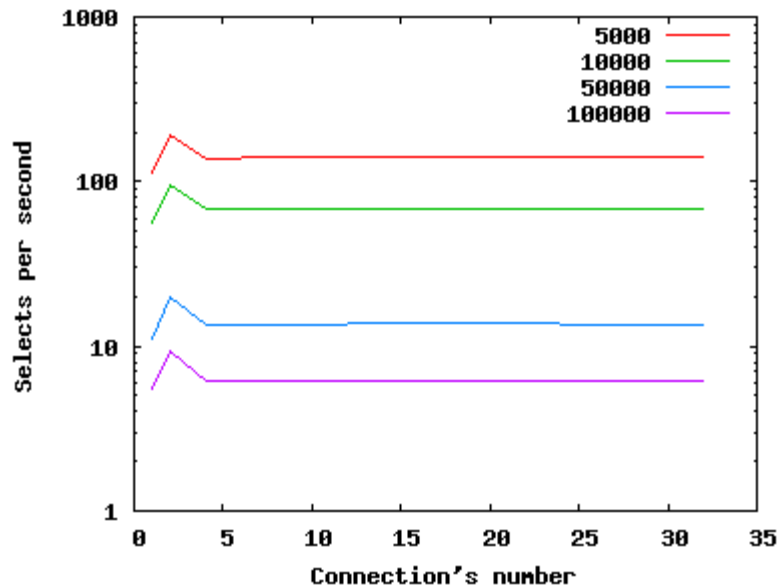
- Размер таблицы учитывает `pg_toast` и `pg_toast_index`
- Указано отношение размеров индекса и таблицы

Производительность GIN



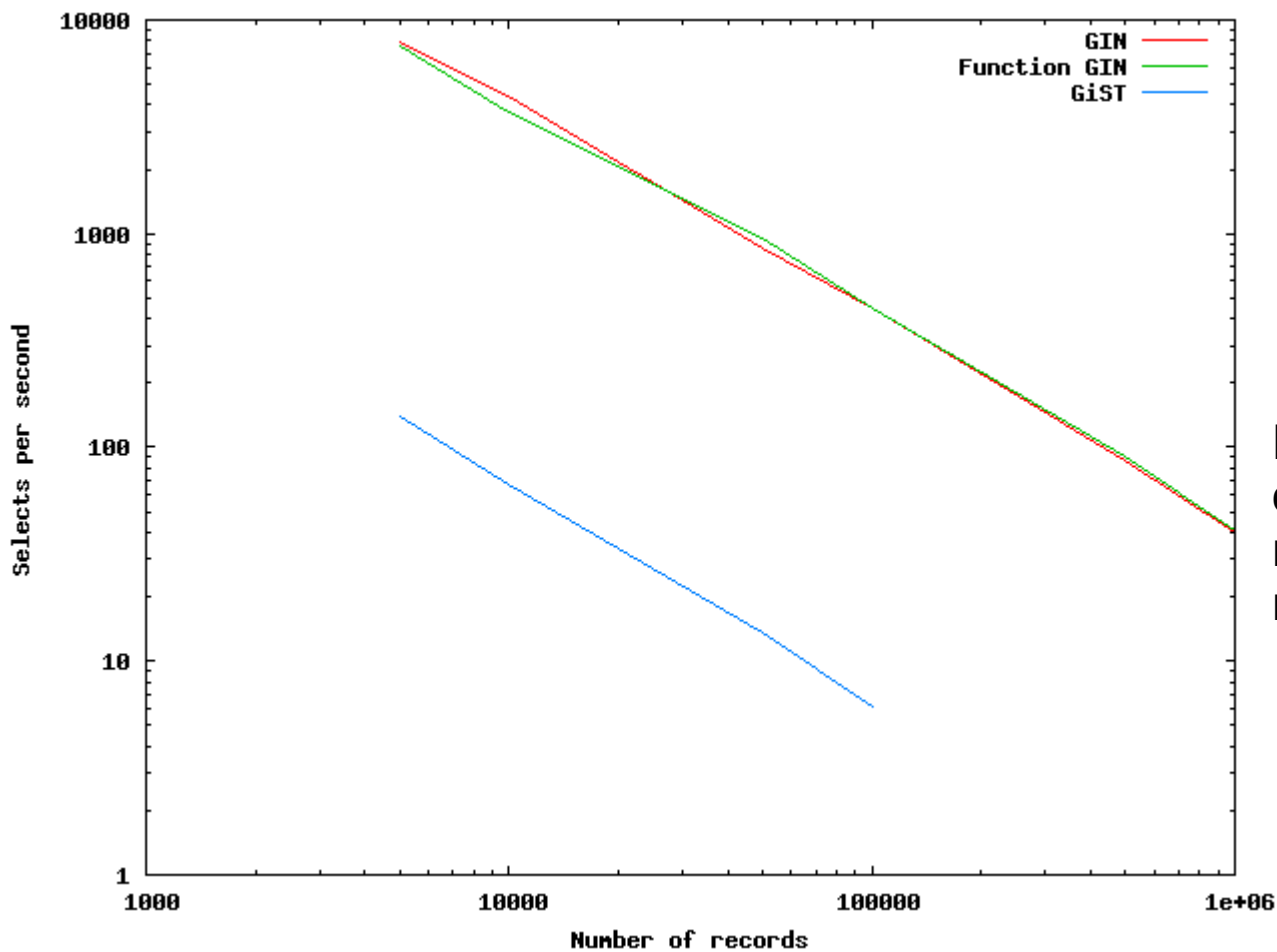
- Увеличение кол-ва процессов практически не влияет на производительность
- Пока база помещается в памяти, кол-во вставок постоянно

Производительность GiST



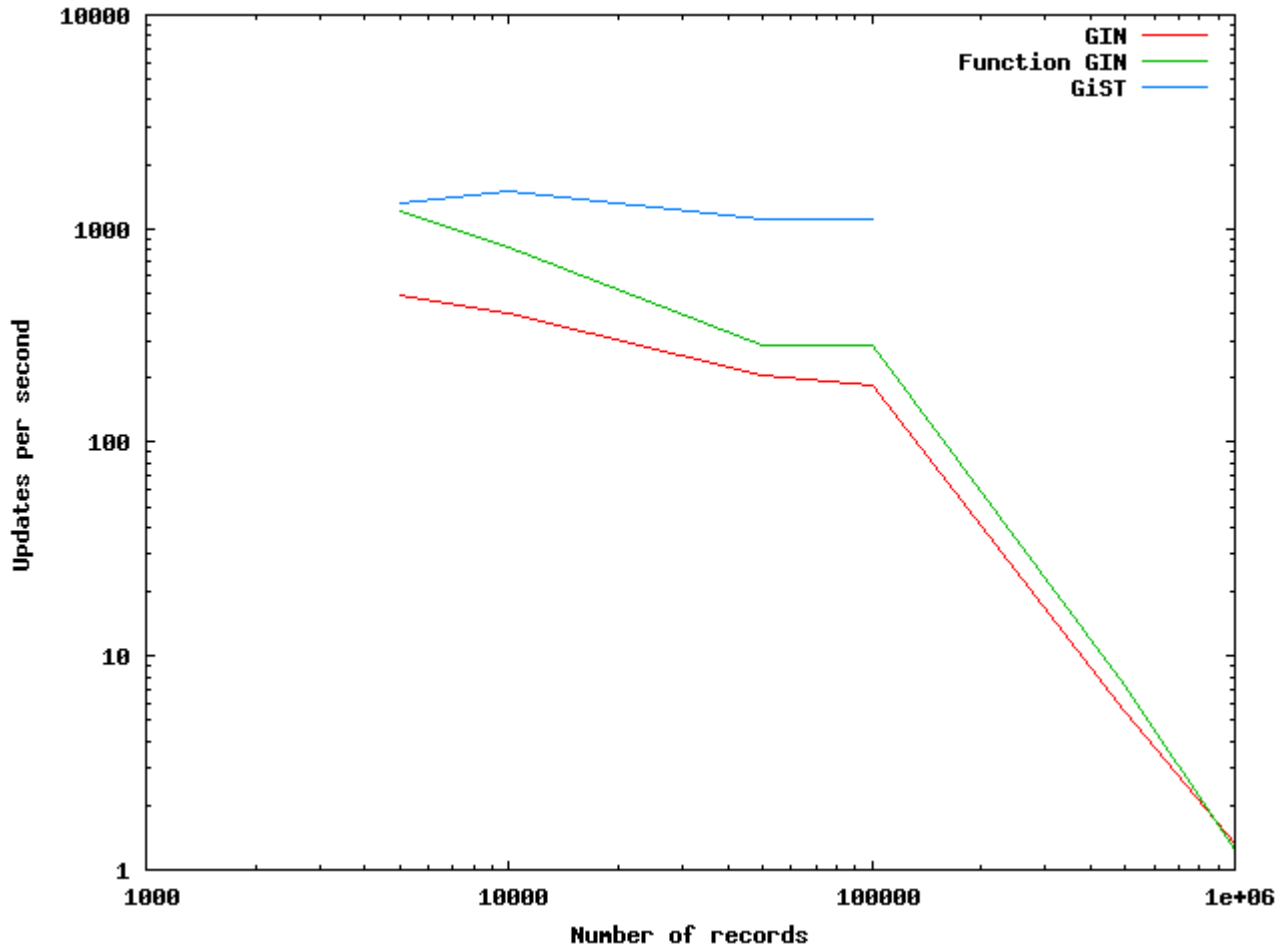
- Увеличение кол-ва поцессов практически не влияет на производительность
- Производительность вставки мало зависит от исследуемых параметров

GIN и GiST на выборке



Производительность
обоих индексов обратно
пропорциональна
количеству записей.

GIN и GiST при обновлении



Производительность GIN при вставке быстро падает из-за большого кол-ва записей в индексе



Выводы

- GIN имеет более высокую производительность при поиске (может выполнить порядка 10 миллионов запросов)
- GiST быстрее при вставке
- В нагруженных системах нужно использовать комбинацию индексов

