

Использование префиксных деревьев для организации индексов баз данных

Таранов Илья Сергеевич

ИСП РАН

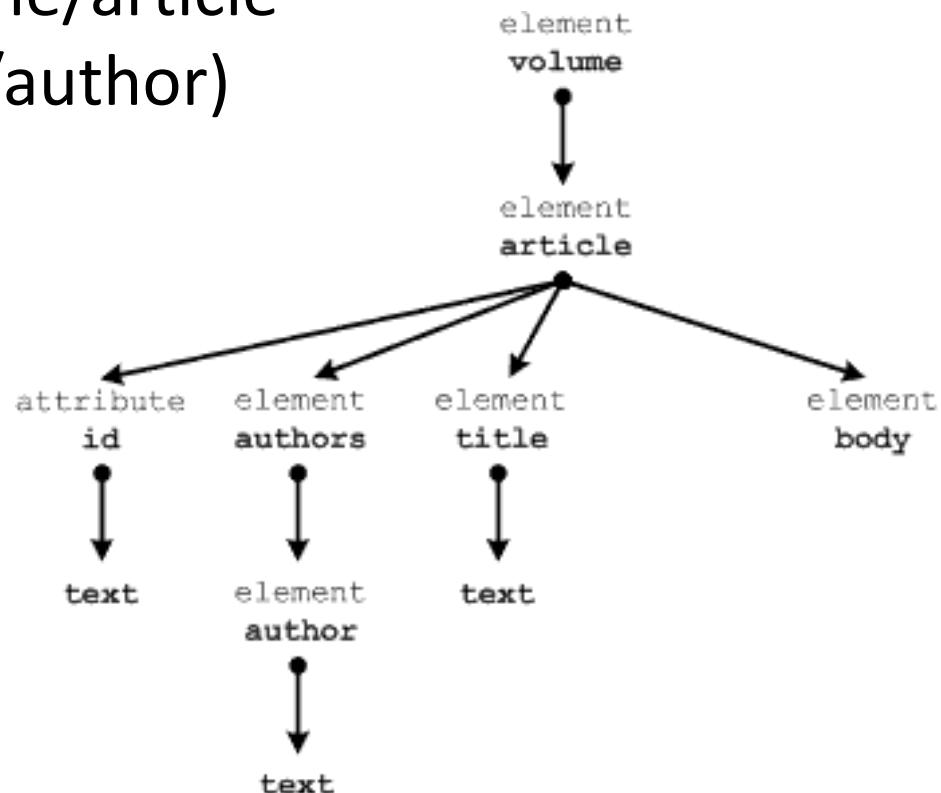
epsilon@ispras.ru

Индексы в XML-СУБД

- Преимущественно текстовые ключи
- Ключ может быть вычислимым выражением
- Возможны дубликаты пар (ключ, значение)

Запрос для создания индекса

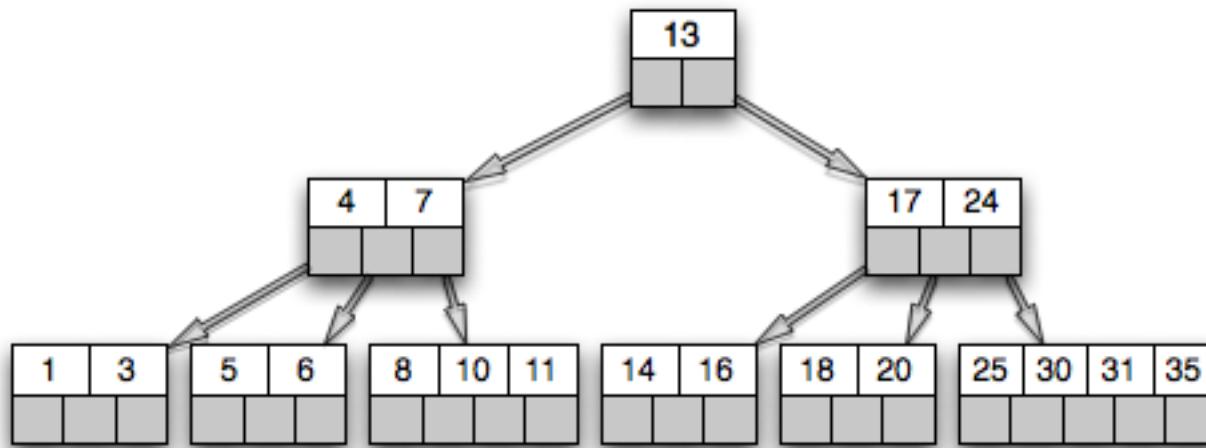
```
CREATE INDEX “index”
  on doc(“a”)/volume/article
  by string-value(.//author)
  as xs:string
```



Методы индексации

- B⁺-tree
- Prefix B-tree
- String B-tree

B⁺-tree



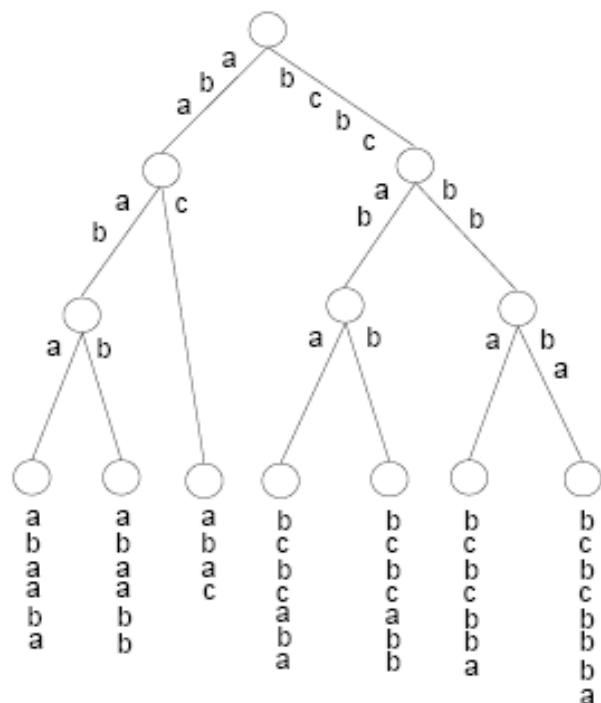
B⁺-tree

- Достоинства:
 - Низкая высота за счёт сильной ветвистости
 - Заполнение страниц минимум наполовину
 - Сбалансировано по построению
 - Поддержка порядка строк (можно выполнять поиск диапазона значений)
 - Ключи хранятся в дереве
- Недостаток:
 - Неудобство работы со строковыми ключами произвольной длины.

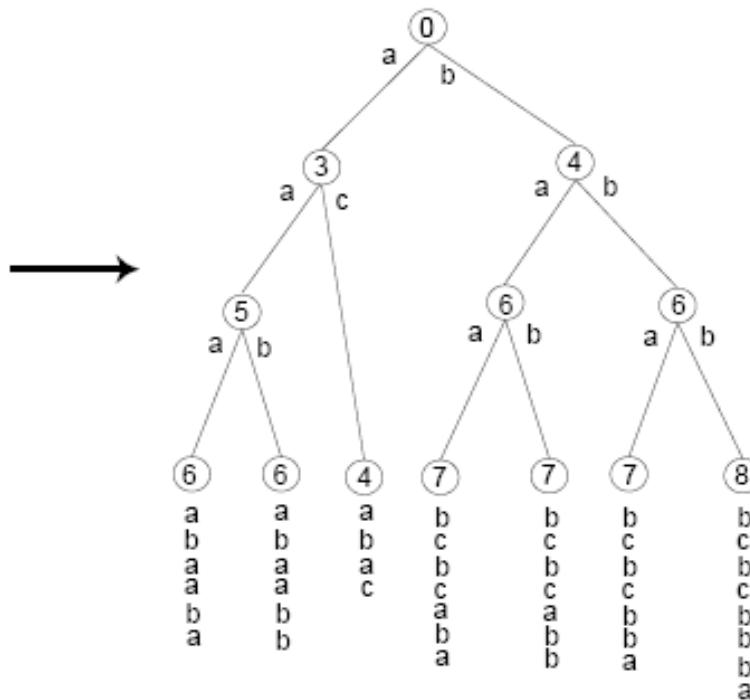
Prefix B-tree

- В узловых вершинах хранятся не строки, а минимальный общий префикс последнего ключа левой страницы и первого ключа правой.
- Недостаток:
 - В худшем случае не лучше B⁺-tree

Trie (Бор) и Patricia

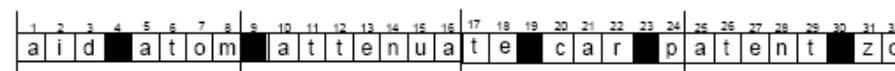
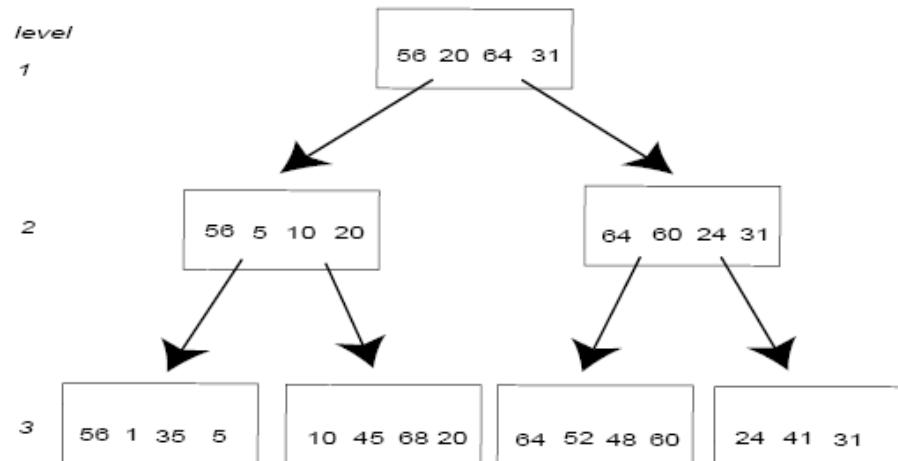


Compacted Trie



Patricia Trie

String B-tree



String B-tree

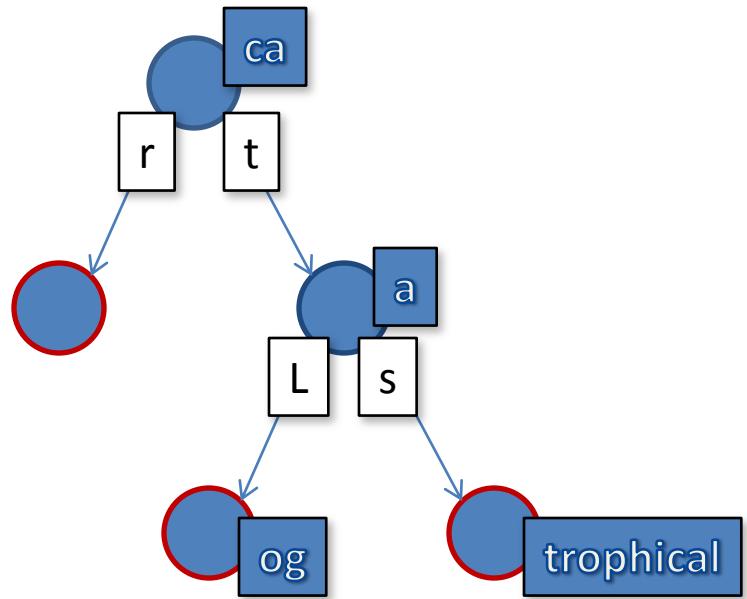
- Достоинства
 - Построение полнотекстовых индексов
 - Отсутствие ограничения на длину строки
 - Возможность поиска подстроки
- Недостатки:
 - Ссылки могут быть достаточно большими
 - Дерево не содержит ключей (для сравнения ключей необходимо обращение ко внешней памяти)

Trie для внешней памяти

- Локальность добавления строк (изменения затрагивают не более 3 страниц)
- Низкая избыточность (не хранятся указатели на отсутствующие поддеревья)
- Хранение ключей

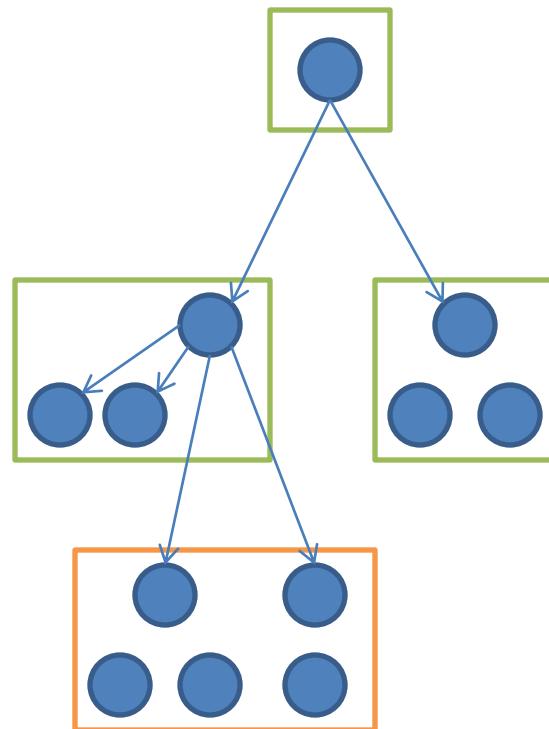
Узел

- Узел дерева S содержит:
 - общий префикс своих потомков
 - список символов перехода
 - список указателей на потомков



Страница узлов

- Страница может содержать одно или несколько поддеревьев
- Ссылка на узел может быть
 - короткой (если узел содержится внутри страницы) или
 - длинной (узел содержится на другой странице)
- Если страница содержит несколько поддеревьев, у всех них общий префикс



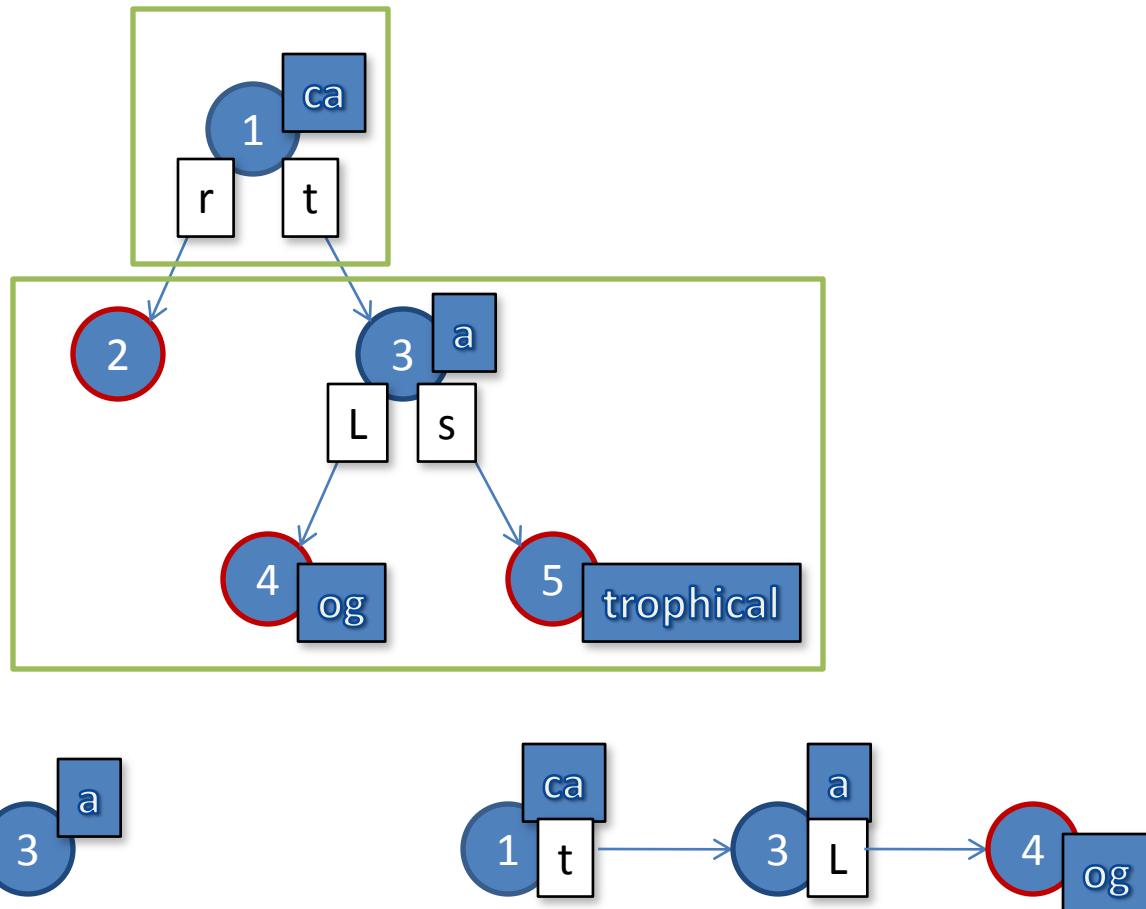
Поиск (1)

- Пусть есть строка Q и узел S_i , а так же список найденных узлов (путь к строке) T .
- Если строка Q пустая, то считаем, что поиск завершён. Поиск считается удачным, если узел S_i является конечным (стоит флаг $F(S_i)$).
- Если префикс $P(S_i)$ не является префиксом Q , считаем поиск неудачным.
- Если префиксы $P(S_i)$ и Q совпадают, то ищем в $M(S_i)$ первый символ строки Q .
- Если такой символ c_j найден, добавляем в T тройку $(S_i, j, B(S_i))$, где B – страница, в которой находится S_i
- Если поиск не завершён, результатом поиска считается строка Q' , такая что $Q = P(S_i) + c_j + Q'$

Поиск (2)

- Пусть есть строка Q , пустой список T , а также корневой узел S_0 (будем считать, что его префикс уже удалён).
- Ищем в $M(S_0)$ первый символ строки Q (c_j).
- Если он не найден, то заканчиваем поиск, поиск считается неудачным. Добавляем $(S_0, \text{null}, B(S_0))$ в T .
- Если он найден, добавляем $(S_0, j, B(S_0))$ в T . Получаем строку Q' : $Q = c_j + Q'$
- Повторяем предыдущую процедуру поиска для (Q', S_i, T) , получая каждый раз остаток строки Q'_{n_i} , пока поиск не будет завершён.

Поиск (илюстрация)



Добавление

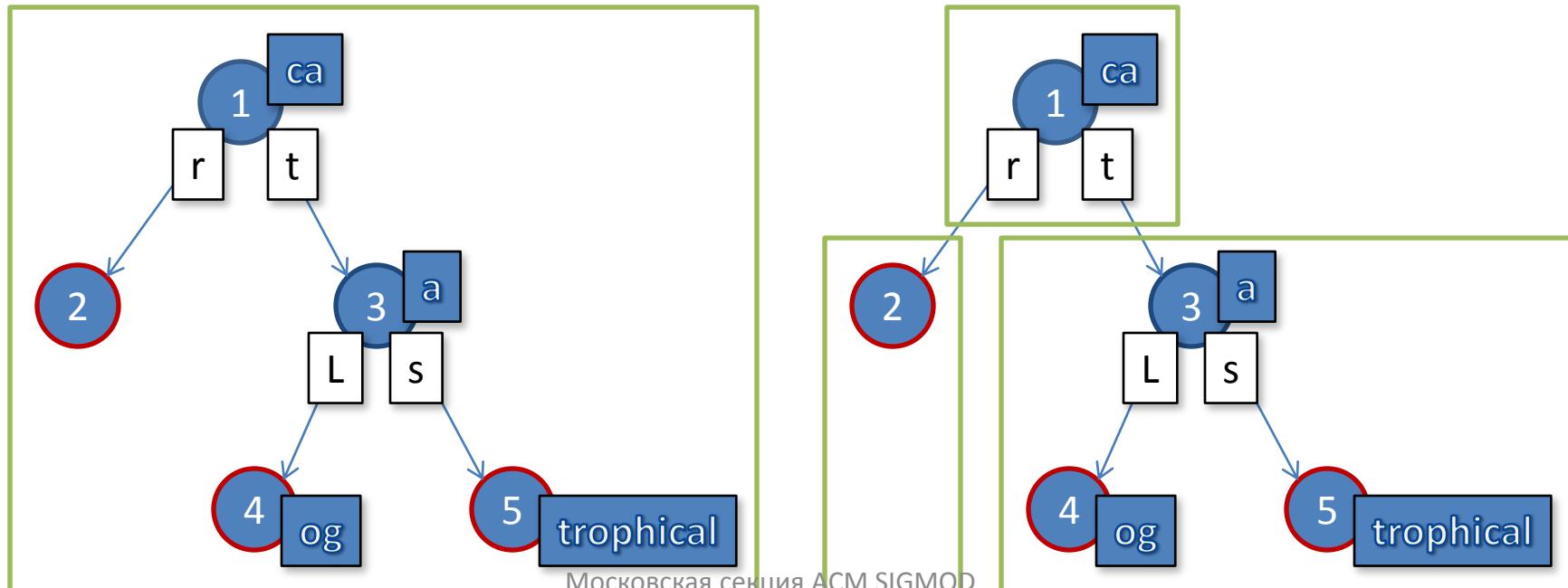
- Добавление строки Q всегда начинается с поиска строки в дереве с получением пути T и остатка строки Q' .
- После поиска мы знаем страницу B , в которую строка должна быть добавлена, а так же узел S , который должен быть для этого изменён.
- Если в B достаточно места, добавляем Q' к S стандартным для префиксных деревьев способом.
- Если места недостаточно, то происходит расщепление страницы

Расщепление страницы

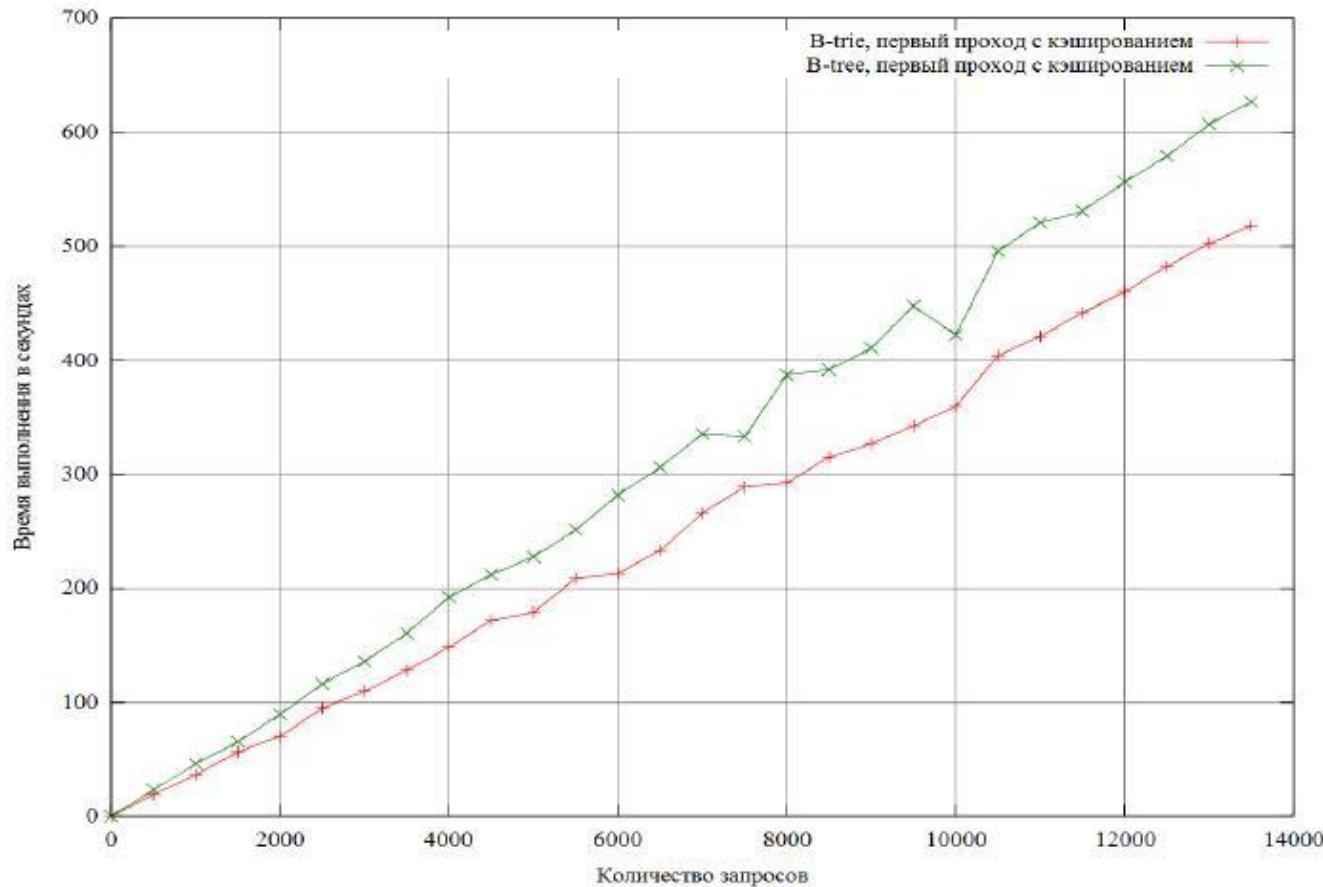
- Если в странице более одного поддерева, то с помощью жадного алгоритма набираем поддеревья, пока они не займут хотя бы половину страницы.
- Набранные поддеревья выделяем в отдельную страницу.

Расщепление страницы

- Если в странице ровно одно поддерево, то выделяем его корневой узел в отдельную страницу, а оставшиеся поддеревья расщепляем предыдущим алгоритмом.



Сравнение с B-tree



Сравнение с B-tree

