КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

УДК 004.651.5

ПОСТРОЕНИЕ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ИНДЕКСИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ФРАКТАЛЬНЫХ ДЕРЕВЬЕВ

К. В. КУЗНЕЦОВ
В. М. МИХЕЛЕВ

Белгородский государственный национальный исследовательский университет

e-mail:
chapaev28@ya.ru
mikhelev@bsu.edu.ru

Исследование процессов построения индексов. Создание новых структур данных с использованием фрактальных деревьев. Проведение вычислительных экспериментов по распараллеливанию операций вставки в индекс.

Ключевые слова: индексирование, построение индексов, B+деревья, фрактальные деревья.

С каждым годом растёт объем информации, который необходимо хранить. Главной задачей, связанной с хранением информации, является её поиск, что особенно актуально в высокопроизводительных компьютерных системах [1]. Для реализации быстрой операции поиска необходимо создавать дополнительные структуры данных, называемые индексами. По способу построения структуры индексов их можно разделить на две группы. Первая группа используется для доступа к обычным данным, таким как строки и данные численного типа, а вторая – для хранения геометрических примитивов, таких как точки и полигоны. Такое разделение определяет алгоритмы доступа к индексам. Например, запросы на вхождение числа в диапазон (range query) или запрос на вхождение в многоугольник. В этой статье рассматриваются вопросы построения индексов, созданных для поиска обычных типов данных.

Во многих современных СУБД для построения индексов часто используют такую структуру данных, как B-дерево. Существует много разновидностей B-деревьев, одна из наиболее используемых это B+деревья. Отличием этой структуры данных является, то что все данные ключей содержатся только в листьях.

На рис. 1 представлена структура B+ дерева (b+tree). Два листовых узла этой структуры данных заполнены не до конца, что отражается на скорости поиска. Также существуют 2-3 деревья, в которых дается гарантия, того что все узлы будут заполнены на две третьих. Основные проблемы использования B-деревьев проявляются при работе с жестким диском [2].

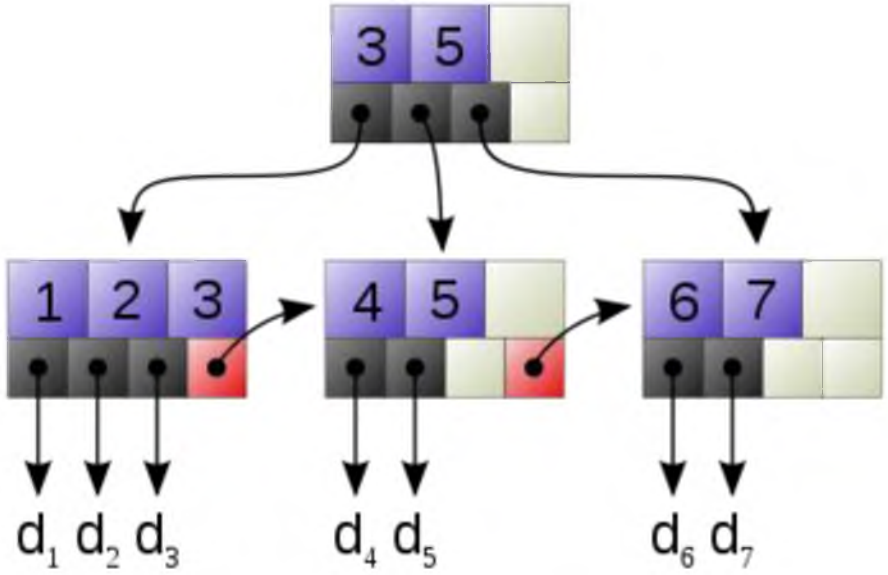

Рис. 1. Схема стандартного B+дерева

В наше время в современных СУБД в качестве индексов стали применять фрактальные деревья (fractal tree index) (fti). Эта структура представляет собой набор массивов. Длины массивов равны степеням двойки. Важным ограничением при построении такого индекса является то, что подмассив этой структуры может быть заполнен данными целиком, либо быть пустым.

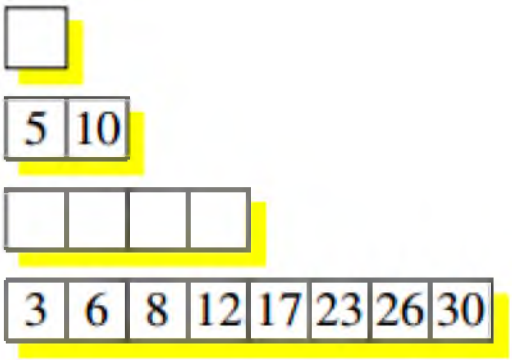

Рис. 2. Схема стандартного fractal tree index

На рисунке 2 показана структура фрактального дерева, состоящего из 4-х подмассивов. Такое представление данных позволяет ускорить процесс вставки в индекс. Допустим, что первый подмассив пуст, тогда вставка происходит в него, если он заполнен, то происходит слияние двух одноэлементных массивов (первый массив – подмассив структуры данных, второй – представляет собой вставляемый элемент, как массив). Результат слияния складывается в массив с длиной равной сумме длин двух подмассивов. Существуют ситуации [3, 4] при которых необходимо выполнить несколько таких слияний.

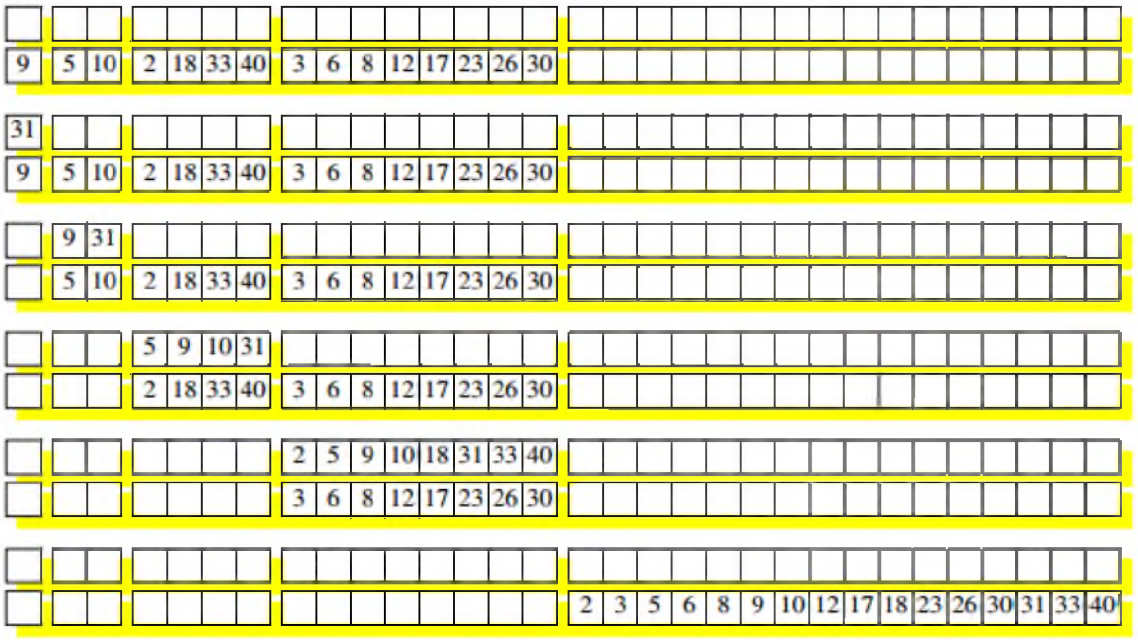

Рис. 3. Схема слияния элементов при вставке

На рис. 3 показан результат, полученный в результате выполнения 4-х слияний. В процессе слияния процедура вставки занимает в лучшем случае 1 операцию, а в худшем – n операций, где n – текущее количество элементов в индексе. Таким образом, сложность операции поиска будет равна log(N)*log(N).

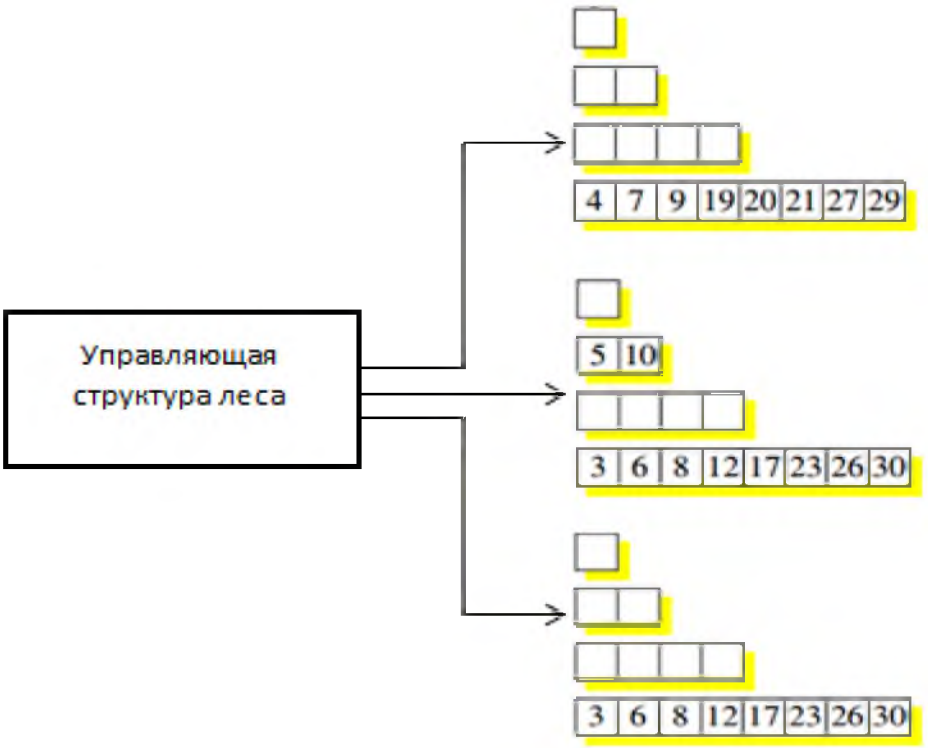

Рис. 4. Схема леса из fractal tree index

Нами предлагается в качестве индекса использовать лес из фрактальных деревьев (forest of fractal tree indexes) (ffti). Такой подход позволит построить эффективный параллельный алгоритм операции вставки.

На рис. 4 представлен лес из 3-х fractal tree index. Выбирается количество деревьев кратное степеням двойки. Под управляющей структурой понимается алгоритм, который будет распределять запросы на деревья леса. Нами предлагается простой для реализации метод, который заключается в том, что необходимо равномерно распределять запросы на вставку по деревьям, а запросы на удаление и выборку необходимо проводить ко всем деревьям.

Далее приведены результаты выполненого вычислительных экпериментов на вставку и поиск для B+дерева, фрактального дерева и леса фрактальных деревьев.

Первый эксперимент – вставка данных. Индексирование проходило по целочисленному ключу. Размер пары ключ–значение равно 16 байтам. Расчет выполнялся для леса из 16 деревьев. На рис. 5 приведены полученные результаты при использовании последовательных алгоритмов вставки.

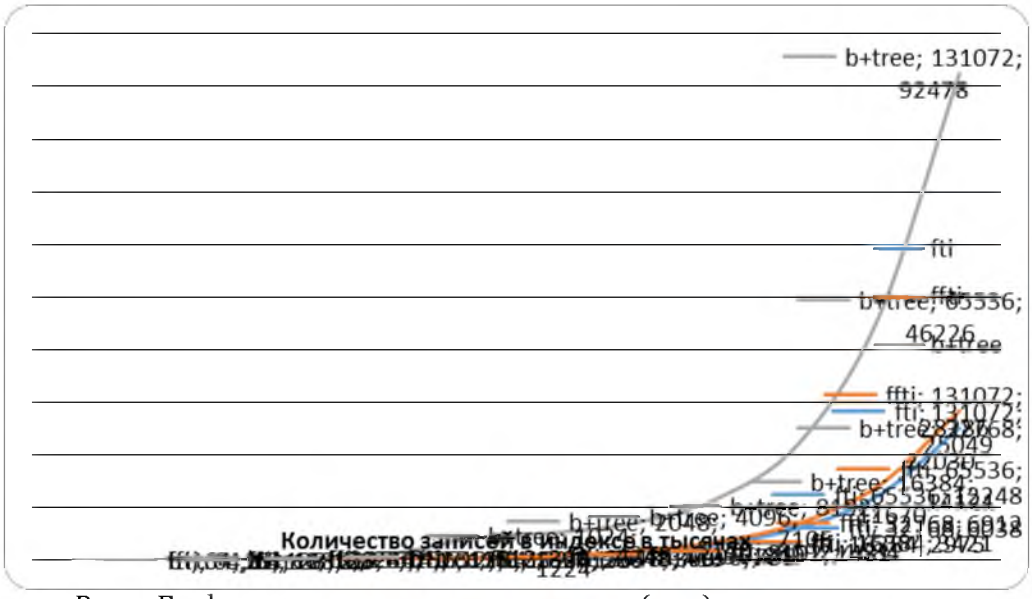

Рис. 5. График зависимости времени вставки (в мс) от размера индекса

Второй эксперимент – исследование скорости поиска. Тип запроса – запрос на вхождение в диапазон (range query). Пример условия 1000<ключ<2000.

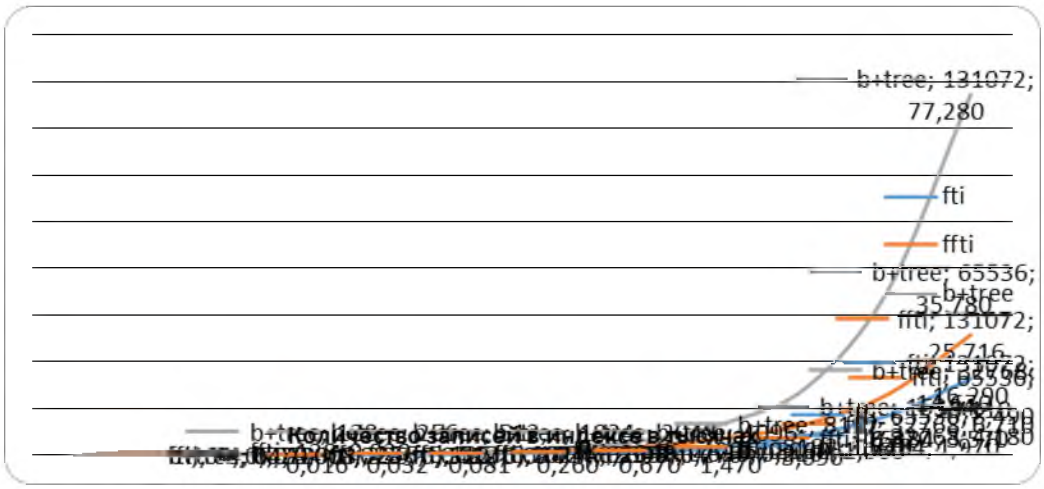

Рис. 6. График зависимости времени поиска (в мс) от размера индекса

На рис. 6 приведены результаты эксперимента по поиску в индексе, в котором использовались только последовательные реализации.

Третий эксперимент – исследование ускорения операции вставки в индекс с использованием технологии параллельного программирования OpenMP.

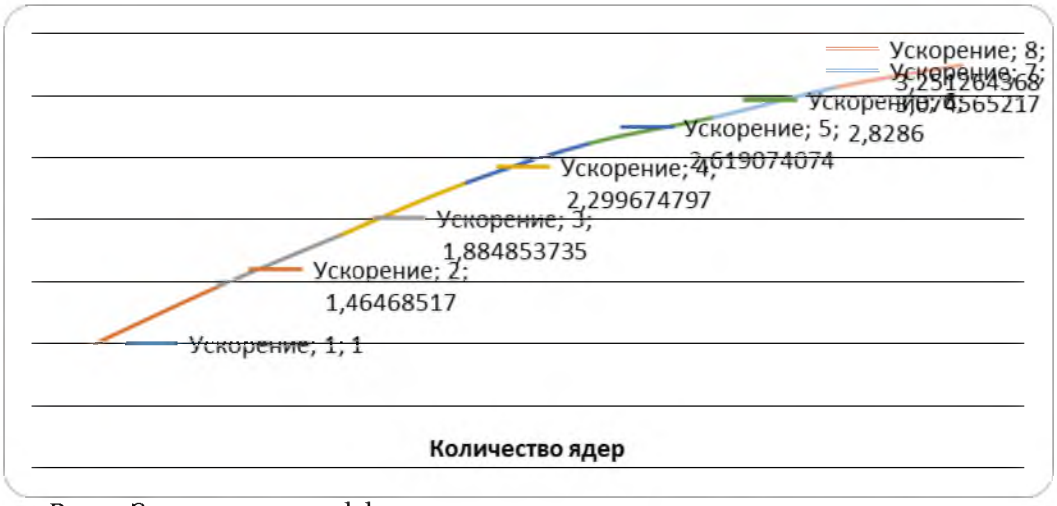

Рис. 7. Зависимость коэффициента ускорения от количества ядер процессора

Количество объектов в индексе 65536 тысяч. Количество деревьев в лесу – 16. Как видно из графика, представленного на рисунке 7, ускорение не пропорционально количеству ядер, но увеличивая количество ядер можно увеличивать и количество деревьев в лесу.

В результате приведенных экспериментов показано, что для ускорения процесса поиска при реализации параллельного алгоритма, необходимо использовать лес фрактальных деревьев при построении индексов. Преимуществом такого подхода является то, что индекс не блокируется полностью, а блокируется лишь его часть – одно дерево. Это свойство предоставляет возможность выполнения параллельной операции вставки и поиска. Предложенный подход также обладает важным свойством масштабируемости, так как лес фрактальных деревьев можно построить из большого количества деревьев, что особенно актуально в эпоху многоядерности современных вычислительных систем.

Список литературы

1. Михелев М.В., Кузнецов К.В., Михелев В.М. Способ построения информационной инфраструктуры высокопроизводительной компьютерной системы для реализации облачных вычислений. [Текст] // Журнал Вопросы радиоэлектроники Серия Электронная вычислительная техника, выпуск 1. Москва, 2012 с. 12-20.

2. Bender, M. A.; Farach-Colton, M.; Fineman, J.; Fogel, Y.; Kuszmaul, B.; Nelson, J. (June 2007). "Cache-Oblivoius streaming B-trees". Proceedings of the 19th Annual ACM Symposium on Parallelism in Algorithms and Architectures (CA: ACM Press): 81–92.

3. Esmet, J.; Bender, M.; Farach-Colton, M.; Kuszmaul, B. (June 2007). "The TokuFS Streaming File System". Proceedings of the 4th USENIX Conference on Hot Topics in Storage and File Systems (MA: USENIX Association): 14–14.

4. Brodal, G.; Fagerberg, R. (Jan 2003). "Lower Bounds for External Memory Dictionaries". Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (N.Y.: ACM Press): 546–554.

PARALLEL ALGORITHM BUILDING INDEX USING FRACTAL TREE INDEXES

C. V. KUZNETSOV
V.M. MIKHELEV

Belgorod State National Research University

e-mail:
chapaev28@ya.ru
mikhelev@bsu.edu.ru

Researching of building index process. Development new data structure using fractal tree indexes. Undertaking of experiment for parallel insertion and searching operations.

Keywords: indexing, building index, fractal tree index, b+trees.