

РЕАЛИЗАЦИЯ ИНДЕКСОВ В СУБД ДЛЯ КЛАСТЕРНЫХ СИСТЕМ

К.В. Бородулин

Одним из важнейших параметров СУБД является время, которое затрачивается для обработки запроса. Индексирование данных позволяет уменьшить время, затрачиваемое для выполнения запроса. В данной работе рассматривается подход к организации индексов в параллельной СУБД «Омега» для кластерных систем.

Для организации эффективного выполнения запросов в работе [1.] был предложен метод частичного зеркалирования, реализованный в СУБД «Омега». В основе данного метода лежит принцип фрагментации данных для распределенного хранения их на различных узлах многопроцессорной иерархии. Метод частичного зеркалирования задает естественный порядок следования кортежей, который определяется порядком считывания кортежей отношения с диска в процессе обработки запроса. При организации индекса по одному или нескольким полям, естественный порядок чтения кортежей нарушается, что делает невозможным использование балансировки загрузки.

Пусть фрагмент F_0 разбит на сегменты для применения алгоритма балансировки загрузки [2.] и на нем определен индекс I_0 . Допустим, что этот фрагмент обрабатывается параллельным агентом Q_0 . Допустим, агент Q_1 закончил обработку данных, выделенных ему, и обращается за новой порцией данных – необработанными кортежами фрагмента F_0 . Так как агент Q_0 использует индекс, а, значит, и процесс считывания кортежей происходит в произвольном порядке, мы не можем определить, какие кортежи уже обработал агент Q_0 , следовательно, возможна ситуация, когда и агент Q_0 и агент Q_1 обработают один и тот же кортеж.

Предположим, что все тот же фрагмент F_0 индексирован по разным атрибутам, таким образом, имеется два индексных файла I_0 и I_1 для одного фрагмента F_0 . И вернемся к ситуации с параллельным агентом Q_0 , у которого более быстрый параллельный агент Q_1 запрашивает порцию данных на обработку. Если в предыдущем примере можно было перестроить фрагмент для восстановления естественного порядка, то при наличии нескольких независимых индексов реструктуризация отношения совершенно невозможна.

Исходя из выше сказанного, был предложен механизм индексации, подходящий для применения в методе частичного зеркалирования. В методе сохраняется горизонтальная фрагментация отношений базы данных. При этом индекс создается сразу по всему отношению. Это означает, что используется фрагментная индексация данных. Так как в таблице индексов содержится поле, содержащее указатель на позицию кортежа в файле базы данных, то воспользуемся информацией, которое несет данное поле, для вычисления номера сегмента, в котором находится данный кортеж. Номер сегмента, в котором находится кортеж, вычисляется по формуле $N=P/S$, где P - позиция кортежа в файле, S - размер сегмента.

Метод состоит в отсеивании выбранных кортежей по признаку нахождения в сегменте. Принцип работы метода:

6. При инициализации индекса, бинарным поиском ищется первый указатель на кортеж, удовлетворяющий нужному условию;
7. Начинается последовательный перебор кортежей, пока атрибут удовлетворяет условию. Для того чтобы сымитировать нормальный порядок следования кортежей, при поиске кортежа в таблице индексов, поиск начинается с первого сегмента.
8. Если номер сегмента кортежа не равен номеру сегмента, в котором происходит поиск, то данный кортеж отбрасывается. В противном случае, кортеж считывается из файла и передается выше по дереву запроса.
9. Если в данном сегменте не осталось подходящих кортежей, то процесс повторяется с шага 2 с увеличенным на 1 значением текущего сегмента.
- 10.

В случае, если номер текущего обрабатываемого сегмента больше, чем число сегментов фрагмента, в котором проводится поиск, то происходит выход из цикла и возвращается специальный EOF-кортеж.

Для внедрения метода индексации в ПСУБД «Омега» необходима подсистема, которая будет инкапсулировать в себе операции с индексами и работой с файловой системой – менеджер данных. На рис 1 приведены варианты использования данной подсистемы». Далее приводится описание каждого варианта использования.

Вариант использования «Инициализировать менеджер» осуществляет подготовку менеджера к использованию. Также инициализируются такие параметры, как условие поиска и необходимое значение атрибута кортежа.



Рис. 1 Варианты использования подсистемы "Менеджер данных".

Вариант использования «Сброс менеджера» выполняет сброс указателей на начальный сегмент и конечный сегмент поиска на нужные значения.

Вариант использования «Выдать кортеж» выполняет поиск и выдачу кортежа, удовлетворяющего условиям поиска, заданным в вариантах *«Инициализировать менеджер»* и *«Сброс менеджера»*.

Рассмотрим структуру подсистемы «Менеджер данных», изображенную на рис. 1. В процессе работы менеджер данных получает команды, определенные вариантами использования выше. Далее, он передает атрибуты поиска (номер отношения, в котором проводить поиск, начальный и конечный сегменты, условие и «литерал» - необходимое значение атрибута поиска) индексатору. При получении команды «Выдать кортеж», менеджер данных вызывает метод «получить кортеж» файлового менеджера, который, в свою очередь, обращается за номером кортежа к индексатору. Тот возвращает номер кортежа, удовлетворяющего условиям поиска, и файловый менеджер, считав данный кортеж, передает его менеджеру ресурсов, который, в свою очередь, передает его как результат операции «Выдать кортеж». В случае, если необходимых кортежей нет, то менеджер данных возвращает специальный EOF-кортеж.

Для интеграции подсистемы «менеджер данных» в СУБД «Омега» было необходимо внести ряд изменений и модифицировать имеющийся исходный код. Изменения коснулись подсистемы компилятора запросов, в котором во время разбора запроса при обнаружении операции выборки не из промежуточного результата, а из отношения задаются необходимые параметры (условие поиска кортежей и номер атрибута) для менеджера индексов. Наибольшим изменениям подверглась подсистема, организующая поток гранул данных. Так как раньше поток сам, непосредственно, обращался к системе управления файлами, то сейчас, с появлением менеджера индексов, такая необходимость пропала. Также, изменениям подверглась подсистема, отвечающая за реализацию скобочного шаблона JOIN_MainMem, которая производит операцию соединения в оперативной памяти. В реализации СУБД «Омега» работа алгоритма была основана на загрузке одного отношения в ОП и последующем циклическом просмотре его кортежей на возможность соединения с кортежами другого отношения, которые последовательно считываются из файла базы данных. В результате встраивания подсистемы «менеджер данных» в шаблон происходит загрузка только индекса, а не целого отношения. Данный метод можно использовать только в том случае, если отношение индексировано только по атрибуту соединения, в противном случае алгоритм работает неправильно из-за потери кортежей.

Параллельная СУБД «Омега» предполагает работу на кластерных системах, поэтому для тестирования были использованы вычислительные узлы кластера «СКИФ Урал» [3.], установленного в Южно-Уральском государственном университете.

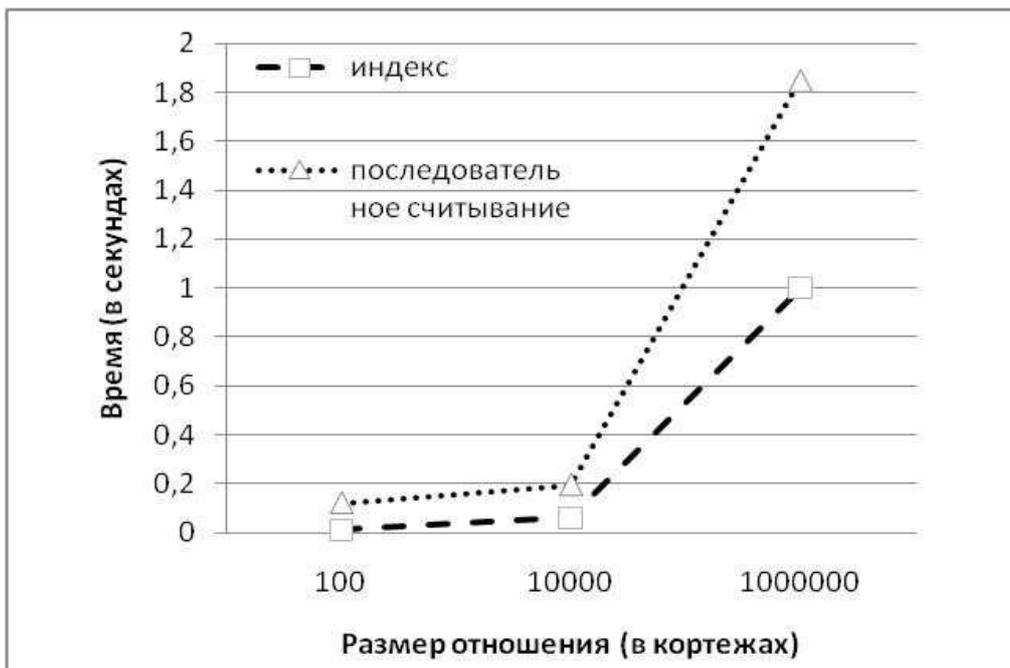


Рис. 2 Время выполнения операции выборки на 2 узлах.

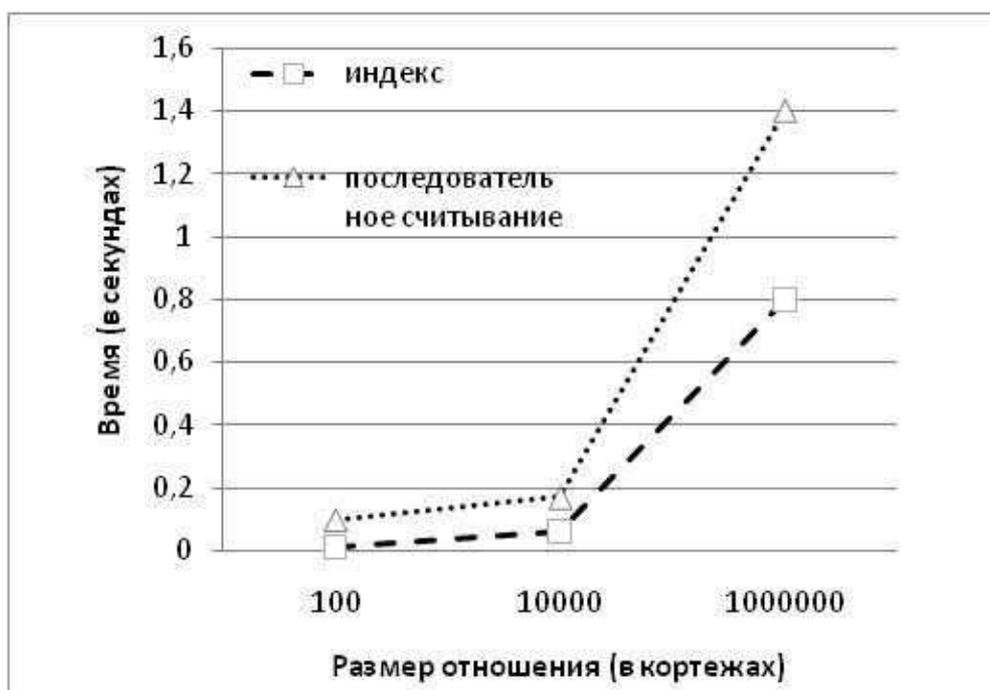


Рис. 3 Время выполнения операции выборки на 4 узлах.

На рис. 2 — рис. 3 показаны графики выполнения операции выборки на 2-х и 4-х вычислительных узлах. Из анализа графиков видно, что при росте размера отношения время выполнения запроса при использовании индексов увеличивается много медленней, чем при последовательном считывании всех кортежей. Также, можно отметить, что при малом размере отношения время, полученное при использовании индексов и без них, различается незначительно.

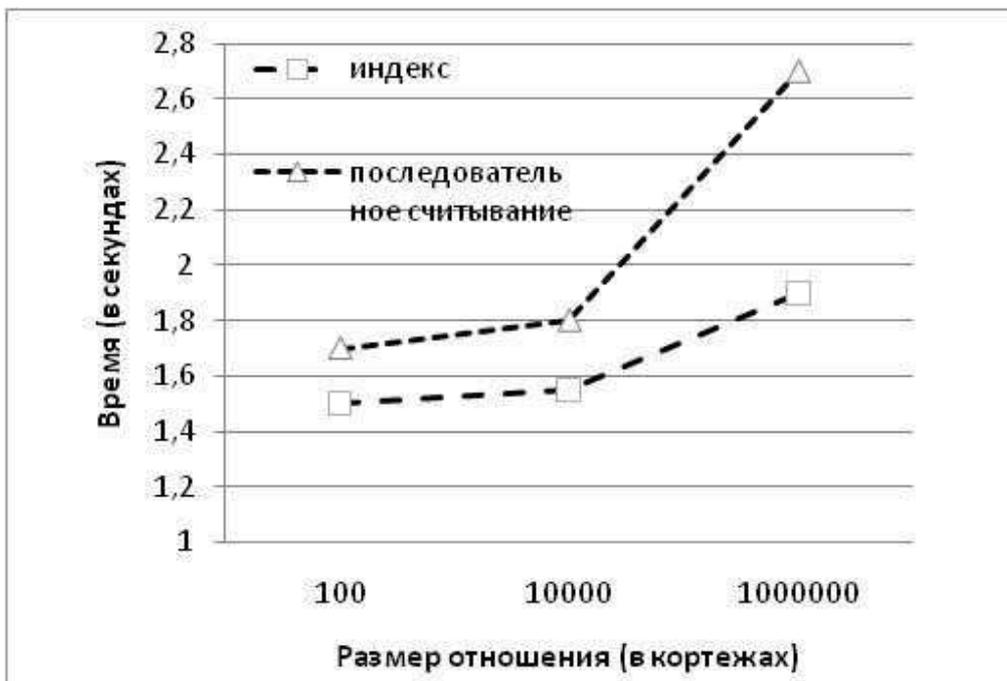


Рис. 4 Время выполнения операции соединения на 2 узлах.

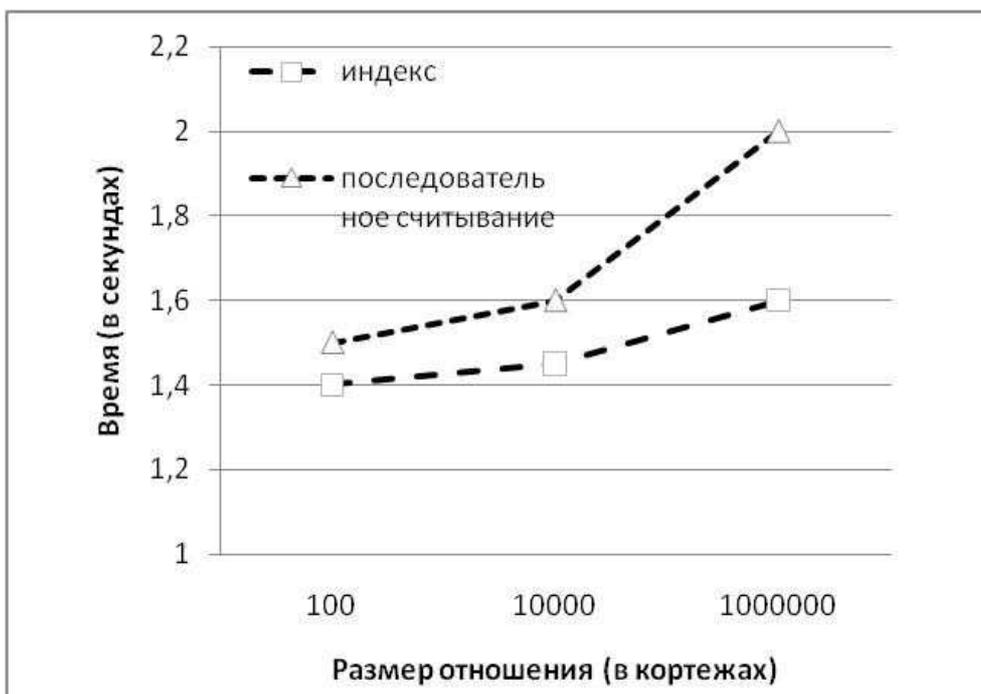


Рис. 5 Время выполнения операции соединения на 4 узлах.

На рис. 4 — рис. 5 показаны графики, демонстрирующие время обработки запроса на разном количестве узлов. Запрос на языке RQL, применяемом в СУБД «Омега»: «J 2 #0 #1», т.е. «Соединить по атрибуту №2 отношения 0 и 1». Отношение 0 фрагментировано по атрибуту соединения. Так как в операции соединения в главной памяти с использованием индекса используется только индекс левого отношения, то размер правого отношения будет одинаково влиять на время запроса, как при использовании индексов, так и без них. Поэтому размер правого отношения был зафиксирован и равен 10000 кортежей. Эксперименты показывают, что использование индексов позволяет существенно сократить время запроса, особенно на отношении с большим количеством кортежей.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 09-07-00241-а).

ЛИТЕРАТУРА:

1. Лепихов А.В., Соколинский Л.Б. Обработка запросов в СУБД для кластерных систем // Программирование. 2010. № 4.
2. Соколинский Л.Б. Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической архитектурой // Программирование. 2001. №. 6. С. 13-29.
3. Высокпроизводительный вычислительный кластер «СКИФ Урал». http://supercomputer.susu.ru/computers/ckif_ural/ (дата обращения 10.04.2010).