УДК 004

Азарян Магдалина Андраниковна,
студент магистратуры
1 курс, институт Цифрового развития
Северо-Кавказского федерального университета
Россия, г. Ставрополь

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО ВЫПОЛНЕНИЯ ОПЕРАЦИЙ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Аннотация: *В статье рассматривается реализация операций реляционных баз данных на параллельных системах с интерфейсом передачи сообщений. Описывается использование различных алгоритмов и суть их работы.*

Ключевые слова: *кортеж, селекция, проекция, операция агрегации, параллельные системы, множества, соединение, трансляционная передача отношений.*

Annotation: *The article discusses the implementation of relational database operations on parallel systems with a message passing interface. It describes the use of various algorithms and the essence of their work.*

Key words: *tuple, selection, projection, aggregation operation, parallel systems, sets, connection, translational transmission of relations.*

Действия над отношениями базы данных (БД) могут быть разделены на два типа: не требующие участия всех кортежей отношений (uniscan) и требующие участия всех кортежей (multiscan). Например, ***соединение*** и ***проекция*** - это mulliscan операции, так как каждый кортеж при их исполнении сравнивается с множеством кортежей. Однако ***селекция*** и ***операция агрегации***

- это uniscan операции, потому что обработка каждого кортежа не зависит от обработки других кортежей [2].

Реализация операций, не требующих участия всех кортежей

В ***селекции*** на горизонтально фрагментированном отношении принимают участие распределенные по вычислительным модулям (ВМ) кортежи. Параллельное исполнение ***селекции*** заключается в чтении каждой ВМ своего резидентного множества кортежей и сравнения для каждого кортежа значения атрибута кортежа с требуемым значением. Если ВМ обнаруживает, что значение атрибута очередного кортежа удовлетворяет оператору ***селекции,*** то она сохраняет кортеж (ссылку па кортеж).

В параллельных системах операции агрегации обычно выполняются за две фазы. На первой фазе каждая ВМ вычисляет свое локальное значение. Кортежи считываются аналогично тому, как в операции ***селекции.*** На второй фазе глобальное значение вычисляется с использованием всех локальных значений и помещается либо в определенную ВМ, либо в управляющую ЭВМ [1].

Для выполнения операции агрегации может быть сформирована структура межмодульных связей типа "дерево". В этом случае операция агрегации выполняется следующим образом. На первом шаге листовые ВМ (т.е. не имеющие потомков) передают свои локально вычисленные значения операции агрегации ВМ-родителям. На последующих шагах ВМ, принявшие сообщение от ВМ-потомков, вычисляют новое значение агрегации с учетом своих резидентных кортежей и передают его вверх по дереву ВМ-родителям Этот процесс продолжается до тех пор, пока результат не достигнет корня дерева. Число шагов пересылки в таком алгоритме будет равно длине самой длинной ветви дерева.

Иногда пользователь может пожелать вычислить какое-нибудь значение по категориям. Например, в БД населения страны пользователь может получить средний возраст населения каждого региона. В этом случае

необходимо произвести перераспределение данных между ВМ так, чтобы в каждой ВМ находились данные, относящиеся к данной категории, затем в каждой ВМ вычисляется свое значение для резидентной категории.

Реализация операций, требующих участия всех кортежей

Параллельные алгоритмы, осуществляющие multiscan операции, могут быть разделены на основанные на трансляционной передаче отношений (broadcast) и основанные на блочной передаче отношений (bucket). Первые требуют, чтобы каждая ВМ посылала фрагменты одного из отношений (в случае двухместного оператора) всем ВМ системы. Все ВМ принимают посланные сообщения (множества кортежей) и выполняют требуемые локальные вычисления с резидентными кортежами и с присланными в сообщениях кортежами.

Алгоритмы, основанные на блочной передаче отношений, используют разбиение всех кортежей на блоки. Каждому блоку соответствует диапазон значений атрибутов, и только кортежи, значения атрибутов которых лежат внутри данного диапазона, находятся в блоке. Блочный алгоритм включает в себя операции внутри блоков согласно значениям их атрибутов.

Соединение по алгоритму, основанному на трансляционной передаче отношения

Рассмотрим пример, поясняющий суть алгоритма. Выполним ***соединение*** двух отношений P и T на атрибуте B, R[ABCD]=P[ABC] (S> T[BD], на системе из трех ВМ при фрагментации отношений, показанной на рисунке 1.

На рисунке 1 отношение P показано как верхнее отношение. ВМ1 имеет 4 кортежа, два P и два T. ВМ2 тоже имеет 2 кортежа из P и два кортежа из T. ВМ3 имеет только 3 кортежа T. Отношение P состоит из меньшего числа кортежей, следовательно, это меньшее отношение.

Алгоритм ***соединения***, основанный на трансляционной передаче отношений, выполняется следующим образом. Исходя из того, что **P** -

меньшее отношение, каждая ВМ посылает каждый кортеж Р во все другие ВМ. Таким образом, ВМ1 посылает кортежи <1,6,4> и <4,9,2>, ВМ2 посылает кортежи <2,6,4> и <3,4,1>, и ВМ3 не посылает никаких кортежей. Конец пересылки кортежей обозначается посылкой стандартного сообщения. Все ВМ отслеживают передачу данных и вычисляют локальное ***соединение*** из присланных кортежей и имеющихся в ВМ кортежей из Т. Рисунок 2 показывает результат ***соединения.***

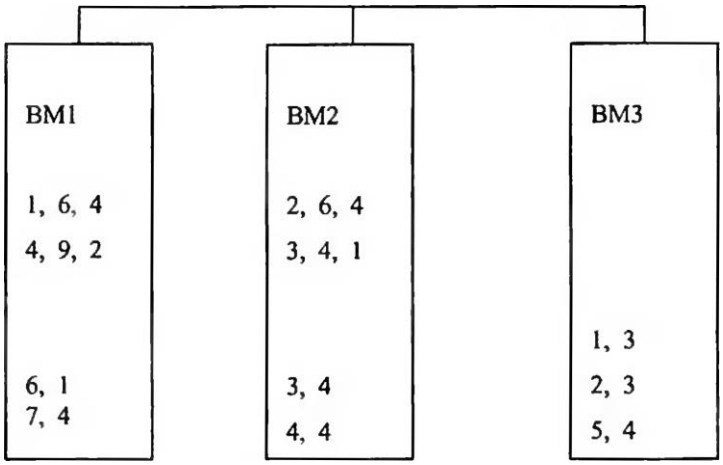

Рисунок 1. Исходная фрагментация отношений

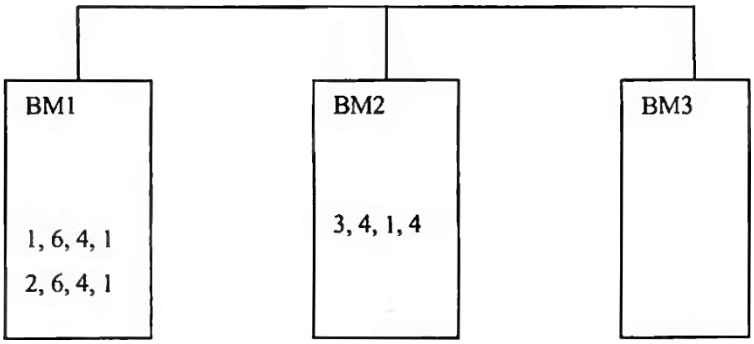

Рисунок 2. Результат соединения по алгоритму, основанному на трансляционной передаче отношения

Алгоритм, основанный на трансляционной передаче кортежей отношения, уменьшает объем передаваемых кортежей за счет избыточных сравнений, что потенциально обеспечивает загрузку всех ВМ. Для повышения

загрузки всех ВМ работой в [4] предложена трехстадийная реализация *соединения.* Первая стадия реализует динамический алгоритм перераспределения данных и имеет целью равномерное распределение исходных кортежей по ВМ [3]. При этом, с одной стороны, увеличивается время перераспределения данных, но, с другой стороны (приблизительно на 35%), уменьшается общее время выполнения операции ***соединения.***

На второй стадии сжатия и копирования отношений копируется меньшее отношение R1. Цель этого шага - увеличить число кортежей из R1, записанных в каждой ВМ, до тех пор, пока не исчерпается свободная память в ВМ, или до тех пор, пока R1 не будет полностью скопировано во все ВМ.

На третьей стадии происходит собственно выполнение операции.

Использованные источники:

1. Корнеев В.В. Параллельные вычислительные системы / В.В.Корнеев – М.: Нолидж, 1999. – 320 с
2. Дэвид Девитт. Параллельные системы баз данных: будущее высоко эффективных систем баз данных. Системы Управления Базами Данных / Девитт Дэвид //Системы Управления Базами Данных – 1995, –№ 2, 8-31 с.
3. Соколинский, Л.Б. Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической архитектурой / Л.Б. Соколинский //Программирование. – 2001, – № 6, – С. 13 - 29.
4. Дейт К. Дж. Введение в системы баз данных— 8-е изд. / К. Дж. Дейт — М.: Вильямс, 2006, 1328 с.