УДК 681.3.06

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ПОКАЗАТЕЛЕЙ СЛОЖНОСТИ ВЫПОЛНЕНИЯ ЗАПРОСОВ В РЕЛЯЦИОННЫХ СУБД

к.т.н. С.С. Танянский, к.т.н. В.А. Филатов, к.т.н. В.В. Тулупов
(представил д.т.н., проф. Е.П. Путятин)

В статье рассмотрены методы обработки запросов в реляционных системах управления базами данных. Основная сложность повышения эффективности работы информационной системы во многом зависит от организации доступа к данным. В качестве формальных средств доступа используется реляционная алгебра. Основываясь на некоторых свойствах операций, даны оценки вычислительной сложности при различных вариантах применении этих операций в запросах.

Постановка проблемы. Одной из важных компонент автоматизированных систем обработки данных является система управления базами данных, обеспечивающая эффективный доступ к данным. Кроме этого современные технологии организации распределенных структур баз данных (БД) должны дополнительно обеспечивать представление данных в виде набора изменяющихся во времени экземпляров отношений, отображающих состояние объектов предметной области и связей между ними, а также поддерживать эффективное выполнение операций для манипулирования этими отношениями.

Анализ предметной области. Можно выделить два направления, решающие вопросы обеспечения высокой производительности информационной системы. Первое направление определяется совершенствованием технической базы, второе – совершенствованием технологий обработки данных. Совершенствование технической базы повлекло развитие многопроцессорных и сетевых технологий, а исследования в области обработки данных определили ряд задач оптимизации поиска информации в БД. Такая организация должна обеспечивать доступ к данным минимально возможное время. При этом оценка времени во многом зависит как от стоимости отдельных операций, так и от совокупности их выполнения.

Поиск наиболее эффективных способов и последовательностей выполнения операций, входящих в конкретный запрос на сегодняшний день, является **актуальной задачей**, требующей дополнительных иссле-

дований и формальных обоснований.

Работы по обеспечению быстрого доступа к данным ведутся с момента разработки языков манипулирования данными. Методы оценивания алгоритмов поиска информации рассматривались в работах Д. Кнута (D. Knuth) [1], А. Ахо (A.Aho) [2], Дж. Хопкрофта (J. Hopkroft) [2]. Практическое использование оценок вычислительной сложности при генерации плана выполнения запроса рассматривались в работах Дж. Ульмана (J. Ullman) [3], К. Дейта (J. Date) [5], С.Д. Кузнецова [6] и др.

Многоэтапность обработки запроса определяется его внутренним представлением, т.е. математическим аппаратом, на основе которого осуществляется доступ и выборка данных. В частности, существуют подходы, связанные с преобразованием запроса к алгебраической форме. Такая форма представления более проста в контексте проблем оптимизации. Представление запроса в алгебраической форме упрощает оценивание операций и выбор наилучшего плана.

Используя свойства операций реляционной алгебры и комбинируя последовательность их выполнения, можно добиваться улучшения показателей скорости и/или размерности промежуточных результатов при выполнении запроса.

Особенно остро вопросы повышения эффективности манипулирования данными стоят при распределенной организации информационной системы. Прежде чем начать сбор фрагментированных данных, желательно знать последовательность соединений отношений БД, которые возможно также фрагментированы и находятся на различных узлах.

В связи с этим **целью данной статьи** является оценка различных последовательностей соединения отношений БД и определение наиболее эффективной относительно мощности промежуточного результата.

Оценка последовательности соединений отношений. Для заданного запроса существует более чем одно алгебраическое представление, причем некоторые из них могут быть "лучше" других. Качество алгебраического выражения определяется исходя из объема затрат, необходимых для его вычисления. При поиске наилучшего выражения будем использовать функцию стоимости, в соответствии с которой вычисляется сумма затрат, необходимых для выполнения запроса.

Рассматривая распределенные системы можно отметить, что общая схема оптимизации запросов содержит два этапа: глобальную оптимизацию, определяющую общий результат и локальную оптимизацию, выполняемую на задействованном узле сети.

Исходной информацией для локальной оптимизации служит алгебраическое выражение, полученное на этапе декомпозиции запроса. Сущность

данного шага заключается в том, чтобы локализовать участвующие в запросе данные, используя информацию об их распределении. При этом выявляются фрагменты, реально участвующие в запросе, а запрос преобразуется к форме, где операции применяются к этим фрагментам. Правила фрагментации обычно выражаются посредством реляционных операций – селекции для горизонтальной фрагментации и проекции для вертикальной.

Сложность оценивания запроса возникает на этапе глобальной оптимизации и сводится к отысканию такой стратегии выполнения запроса, которая была бы в худшем случаи достаточно близкой к оптимальной, так как нахождение оптимальной стратегии, в общем случае, является вычислительно трудноразрешимой задачей [2].

Для того чтобы определить “наилучший” порядок выполнения операций, необходимо оценить вычислительные затраты для множества альтернативных вариантов. Определение вычислительных затрат до выполнения запроса основано на формулах оценки мощности результатов выполнения отдельных операций [4]. Важнейшим аспектом эффективности является порядок выполнения операций соединений, поскольку его изменение может привести к ускорению до нескольких порядков.

В силу того, что операция соединения обладает свойством коммутативности и ассоциативности, последовательность соединений в дереве операций не обязательно должна быть линейной [5].

Таким образом, запрос, требующий соединений отношений R_1, R_2, R_3, R_4, R_5, может быть алгебраически представлен как линейная последовательность $R_1 \triangleright\triangleleft R_2 \triangleright\triangleleft R_3 \triangleright\triangleleft R_4 \triangleright\triangleleft R_5$, либо как парная последовательность операций $(R_1 \triangleright\triangleleft R_2) \triangleright\triangleleft (R_3 \triangleright\triangleleft R_4) \triangleright\triangleleft R_5$.

Рассматривая всевозможные комбинации, можно отметить смешанный вариант соединений, то есть часть отношений соединяются заранее определенными парами, а часть произвольно. Например, последовательность операций может выглядеть как $R_1 \triangleright\triangleleft (R_2 \triangleright\triangleleft R_3) \triangleright\triangleleft R_4 \triangleright\triangleleft R_5$.

В настоящее время большинство систем использует линейные последовательности, хотя с точки зрения минимизации размерностей промежуточных результатов парное соединение имеет ряд преимуществ [5, 7]. Рассмотрим количественную характеристику различных способов соединений отношений. При сравнении будем брать во внимание оценки парной и линейной последовательности выполнения операций.

Пусть БД состоит из множества $R = \{R_1, R_2, \dots, R_n\}$ соединенных между собой отношений, тогда число всевозможных пар для парного соединения определяется как произведение сочетаний из n по 2 без повторений. Так как порядок полученных подмножеств неважен, необходимо исключить все возможные перестановки внутри таких подмно-

жеств, количество которых выражается значением $P_m = m!$, где m – количество пар соединений, $m = \frac{n}{2}$ (округление значения m будем осуществлять к меньшему значению). Таким образом, число всевозможных комбинаций для четного значения n выражается формулой

$$N_{\text{чет}}^{\text{пар}} = \frac{C_n^2 C_{n-2}^2 .. C_{n-2(m-1)}^2}{P_m} = \frac{n!}{(2!)^m m!},$$

для нечетного значения n формула имеет вид

$$N_{\text{нечет}}^{\text{пар}} = \frac{C_n^{n-1}(C_{n-1}^2 C_{n-1-2}^2 .. C_{n-1-2(m-1)}^2)}{P_m} = \frac{C_n^{n-1}(n-1)!}{(2!)^m m!} = \frac{\frac{n!}{(n-1)!}\cdot(n-1)!}{(2!)^m m!} = \frac{n!}{(2!)^m m!}.$$

Таким образом, число всевозможных вариантов парного соединения для любого числа отношений выражается формулой

$$N^{\text{пар}} = \frac{n!}{(2!)^m m!}.$$

При линейном способе соединения количество можно выразить как

$$N^{\text{лин}} = C_n^2 (n-2)! = \frac{n!}{2!(n-2)!}(n-2)! = \frac{n!}{2!}.$$

Очевидно, что при сравнении числа всевозможных пар соединимых отношений парным и линейным способами, выражение будет иметь вид:

$$N^{\text{пар}} < N^{\text{лин}}.$$

Для формального подтверждения проделанных выкладок рассмотрим пример. Пусть $R = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7\}$, т.е. $n = 7$ и $m = 3$, тогда $N^{\text{пар}} = \frac{7!}{(2!)^3 3!} = 105$ и $N^{\text{лин}} = \frac{7!}{2} = 2520$, таким образом, $N^{\text{лин}}$ значительно превышает $N^{\text{пар}}$.

Если ребру дерева присвоить некоторый вес, например мощность отношения промежуточного результата, тогда выбор оптимальной последовательности соединений будет состоять в отыскании для каждого возможного варианта остовного дерева минимального веса.

Основной источник трудностей, возникающий при решении подобной задачи, состоит в том, что вес любого ребра не является величиной постоянной. Причем, с одной стороны, как было показано, вес зависит от последовательности соединений отношений, а с другой стороны, от количества значений соединяемых атрибутов, которое нелинейно возрастает при выполнении операции. Фактически это означает, что вершины соединяются несколькими ребрами с различными весами. Причем все веса по-

прежнему зависят от последовательности соединения отношений БД.

Выводы. Предложенные в статье расчеты позволяют детализировать показатель сложности алгоритмов, как количество сравнимых последовательностей соединений. Хотя действительное значение может быть получено после определения методов соединения, формул расчета затрат и способов получения информации о размерах промежуточных значений.

С другой стороны, полученные оценки показывают, насколько важно использовать эффективные методы обработки данных, хотя в некоторых случаях это может несколько замедлить отклик на запрос. В этом случае можно отметить, что предварительная компиляция программы, выполняемая до ее непосредственного выполнения, оправдывает использования дополнительных алгоритмов при обработке запроса. Таким образом, общий показатель скорости работы информационной системы значительно возрастает при использовании "быстрых" алгоритмов при обработке запросов.

ЛИТЕРАТУРА

1. *Кнут Д. Искусство программирования для ЭВМ. Т. 1. Основные алгоритмы. – М.: Вильямс, 2000. – 520 с.*
2. *Ахо А., Хопкрофт Д. и др. Структуры данных и алгоритмы. – М.: Вильямс, 2000. – 384 с.*
3. *Ульман Дж. Основы систем баз данных. М.: Финансы и статистика, 1983. – 304 с.*
4. *Кормен Т., Лейзерсон Ч. и др. Алгоритмы: построение и анализ. М.: МЦНМО, 2001. – 280 с.*
5. *Дейт К. Введение в системы баз данных. М.: Вильямс, 2001. – 386 с.*
6. *Кузнецов С.Д. Логическая оптимизация запросов в реляционных СУБД // Программирование. – 1989. – № 6. – С. 46 – 59.*
7. *Codd E. A relational model of data for large shared data banks // CACM 13. – 1970. – No. 6. – P. 1958 – 1982.*

Поступила 27.12.2003

***ТАНЯНСКИЙ Сергей Станиславович**, канд. техн. наук, доцент кафедры информационных систем и технологий в деятельности ОВД Национального университета внутренних дел. В 1992 году окончил Харьковский институт радиоэлектроники. Область научных интересов – организация, проектирование и поддержка баз данных.*

***ФИЛАТОВ Валентин Александрович**, канд. техн. наук, доцент кафедры искусственного интеллекта Харьковского Национального университета радиоэлектроники. В 1980 году окончил Харьковский институт радиоэлектроники. Область научных интересов – обработка данных в распределенных неоднородных базах данных.*

***ТУЛУПОВ Владимир Владимирович**, канд. техн. наук, старший преподаватель кафедры информатики Национального университета внутренних дел. В 1991 году окончил Харьковское высшее командно-инженерное училище ракетных войск. Область научных интересов – информационный поиск и организация информации.*