

## ОЦЕНКА ЧИСЛА ДИСКОВЫХ ОПЕРАЦИЙ ОПЕРАТОРОВ ЯЗЫКА МАНИПУЛИРОВАНИЯ ДАННЫМИ В СЕТЕВОЙ БАЗЕ ДАННЫХ

С.В. Осиевский

(представил д.т.н., проф. Г.Ф. Кривуля)

*Предложен способ получения оценки операторов языка манипулирования данными (ЯМД) в целях подготовки исходных данных для построения графика реализации запросов пользователей и транзакций в сетевой базе данных (СБД).*

**Введение.** Для СБД телекоммуникационных сетей (ТС) управление реализацией операторов ЯМД является как проблемой, так и возможностью повышения производительности сети [1]. **Проблема** заключается в том, что некоторые ТС для достижения определенного уровня производительности требуют оптимизации. Оптимизация позволяет улучшить работу ТС, так как применение оптимизации к операторам ЯМД, отображаемых реляционными операторами, переводит их на более высокий семантический уровень. С другой стороны, некоторые ТС предполагают реализацию запросов абонентов на более низком семантическом уровне, в котором любая оптимизация должна выполняться автоматически. В таких системах пользователь, а не система определяет порядок реализации операторов ЯМД [1, 2]. Но независимо от того, кто определяет порядок реализации операторов ЯМД, требуется некоторая оценочная характеристика, позволяющая принять управленческое решение на реализацию оператора в ТС. Все операции, выполняемые в рамках вычисления стоимости реализации операторов ЯМД, являются составной частью вычислительного процесса и могут быть представлены последовательностью процедур, взаимосвязанных информационно или по управлению некоторым эквивалентным алгоритмом [3]. Эта последовательность отображается инструкцией

$$[\theta_{\mu}, \{a_{\mu}\}, \{b_{\mu}\}], \quad (1)$$

где  $\mu$  – код операции, соответствующей данной инструкции;  $\{a_{\mu}\}$  – множество операндов;  $\{b_{\mu}\}$  – множество результатов выполнения инструкции.

Каждая такая инструкция соответствует выполнению отдельной элементарной операции над сетевой базой данных и входит в состав алгоритма решения задачи построения графа допустимых переходов [4]. Выбор порядка следования операторов в алгоритмах построения графика реализации

имеет смысл, если существуют (или их можно получить) оценки используемых ресурсов каждой инструкцией или временем ее выполнения  $t_{ij}$ . Противоречивость существующих способов получения оценки приводит к ограничению возможностей применимости программных средств поставляемых с СУБД для разнородных сетевых баз данных.

**Целью** данной статьи является разработка способа оценки операторов ЯМД подготовки исходных данных для построения графика реализации запросов пользователей и транзакций в СБД.

**1. Формализованное описание процедуры построения матрицы следования.** Введем предположение, заключающееся в том, что совокупность запросов и транзакций при составлении графика реализации представляет собой единый блок с точки зрения определения правил формирования записи на алгоритмическом языке.

При разбиении алгоритма на отдельные процедуры уровень детализации необходимо устанавливать в зависимости от возможностей СУБД по блокированию элементов данных или с учетом максимально допустимого времени выполнения транзакции. При таком разбиении каждый оператор соответствует процедуре, которая впоследствии будет занимать некоторое оптимальное положение в графике реализации.

Пусть множество инструкций-процедур одного линейного участка алгоритма образует множество  $\{0_i, \dots, 0_m\}$ . Для него необходимо построить матрицу следования  $S = \|\alpha_{\mu\nu}\|$  размера  $m$ . Построение матрицы  $S$  как исходной информации для процесса управления реализацией рабочей нагрузки СБД, с помощью алгоритмов, предложенных в [3], позволяет оперативно использовать заключающуюся в ней информацию о стоимости операторов (соответствующих инструкциям-процедурам).

При этом следует учитывать, что назначение и выполнение элементарных операций могут быть совмещены между собой и с составлением матрицы следования. В этом случае матрица следования составляется не для всего линейного участка, а с диспетчером связан буфер памяти объемом  $g$ , в котором формируется текущий вид матрицы следования  $S$ . Динамическое изменение этой матрицы происходит по двум причинам. Во-первых, в результате назначения операторов на процессы из матрицы  $S$  должны быть исключены соответствующие строки и столбцы. Во-вторых, при наличии места в буфере и при необходимости продолжения обзора инструкций-процедур линейного участка должны формироваться новые строки и столбцы  $S$  для продолжения процесса упорядочивания операторов. Поэтому формирование матрицы следования  $S$  производится в результате последовательной обработки операторов так, чтобы размер матрицы  $S$  не превышал  $g$ .

Одновременно по счетчику времени исключаются из матрицы следования строки и столбцы, соответствующие выполненным операторам. Корректируется закрепление строк и столбцов за рассматриваемыми операторами. При выполнении операторов матрица S уплотняется так, что ее строки и столбцы занимают сплошной массив буфера памяти, сконцентрированный в его начале. Числовые значения информации матрицы следования отражают требуемые затраты для реализации оператора ЯМД и могут отображаться либо временем реализации, либо числом дисковых операций, необходимых для реализации оператора ЯМД. Выбор способа оценки операторов ЯМД зависит от алгоритмов построения графика реализации запросов пользователей и транзакций. Рассмотрим способ получения оценки операторов ЯМД для алгоритмов, предложенных в [5, 6].

**2. Выделение подмножества операций реляционной алгебры для получения минимального значения стоимости оператора ЯМД.** В табл. 1 представлено полное множество операций реляционной алгебры над нормализованными отношениями (НО), покрывающее все языковые интерфейсы распространенных СУБД. Особенностью реляционных операций, связанных с интеграцией данных, является возможность порождения значения вычисляемых атрибутов, входящих в схемы результирующих отношений.

Таблица 1

Подмножество операций реляционной алгебры над НО

Сложность	Наименование операции	Система интеграции	Язык SQL	Язык QBE
O(n)	<i>Селекция</i>	+	+	+
	<i>Проекция без удаления дублей</i>	—	—	+
	<i>Конвертирование</i>	+	+	+
	<i>Присваивание</i>	+	+	+
O(n2)	<i>Проекция с удалением дублей</i>	+	+	+
	<i>Объединение</i>	+	+	+
	<i>Пересечение</i>	+	+	+
	<i>Разность</i>	+	+	+
	<i>Соединение</i>	+	+	+
	<i>Соединение по условию</i>	+	+	+
	<i>Ограничение</i>	+	—	—
	<i>Неполное соединение</i>	+	—	—
	<i>Декартово произведение</i>	+	—	—
	<i>Агрегатные функции</i>	+	+	+
	<i>Сравнение</i>	+	+	+
	<i>Сортировка</i>	+	+	+

В соответствии с двухуровневой структурой обработки данных, все реляционные операции обычно разбивают на два класса [1, 2, 7 – 9]. Пер-

*вый класс* составляют операции первичной обработки. Сюда входят операции селекции по условию, проекции отношения без удаления дублей, конвертирования и присваивания. Эти унарные операции являются массовыми и могут быть реализованы как в буферной памяти ЭВМ, так и непосредственно в устройствах массовой памяти (УМП). Поскольку указанные операции выполняются за один просмотр отношения, то сложность их оценивается в  $O(n)$ , где  $n$  – кардинальность отношения-операнда. *Второй класс* составляют такие массовые операции над отношениями, сложность которых оценивается как  $O(n^2)$ . Сюда включаются бинарные операции реляционной алгебры, а также унарные операции проекции с удалением дублей, операции с использованием группировки (group by). С некоторыми допущениями сюда включена также операция сортировки отношений.

С целью детального анализа имеющихся возможностей повышения реактивности СУБД для каждого типа алгебраической операции рассмотрим эквивалентное ей выражение на языке реляционного исчисления. Для обозначения хранимых отношений в алгебре воспользуемся обозначением  $R1(x, y, z, \dots)$ ,  $R2(a, b, c, \dots)$  и т.п., а в исчислении – соответствующие предикаты  $P1(t)$ ,  $P2(t)$  и т.п., где  $t$  – переменная-кортеж [9].

Каждое выражение на языке исчисления будет иметь вид  $\{tp/P\}$ . Эта запись означает множество кортежей  $tp$ , удовлетворяющих логическому условию  $P$ . Здесь  $\langle x, y, z \rangle$  будет означать кортеж с тремя компонентами, тогда как  $t ++ g$  – кортеж, получаемый с помощью присоединения (конкатенации) кортежа  $g$  к концу  $t$ ,  $t - g$  означает кортеж  $t$ , укороченный путем выбрасывания атрибутов, входящих в кортеж  $g$ . Тогда приведенные в таблице операции манипулирования данными формализовано примут вид:

$$\text{селекция: } R1: [f(x) \text{ and } (g(y) \text{ or } \_h(z))] \{t \mid P1(t) \wedge (f(t, x) \wedge (g(t, y) \vee h(t, z)))\}; \quad (2)$$

$$\text{проекция: } R1 \text{ projected\_to\_ } x, y \{ \langle t, x, t, y \rangle \mid P1(t) \}; \quad (3)$$

$$\text{декартово произведение: } R1 ** R2 \_ \{t ++ g \mid P1(t) \wedge P2(t)\}. \quad (4)$$

Аналогичным образом составляются выражения для остальных операций. Полученные результаты являются основными зависимостями, применяемыми для получения соотношений при оценке операторов языка манипулирования данными в терминах числа дисковых операций.

**3. Определение требуемого объема статистической информации.** Поскольку полнота оценки объема и стоимости выполнения операций реляционной алгебры в значительной степени определяется полнотой и актуальностью статистической информации, накапливаемой службами СУБД [1, 2], рассмотрим те данные, которые необходимы для выполняемых расчетов.

*Для каждого базового отношения  $R$ :*

$ntuples(R)$  – количество кортежей (записей) в отношении  $R$  (кардинальность);

$\text{bfactor}(R)$  – показатель блокирования отношения  $R$  (число кортежей отношения  $R$ , размещаемых в одном дисковом блоке);  
 $\text{nblocks}(R)$  – количество блоков вторичной памяти, необходимых для хранения отношения  $R$ .

*Для каждого атрибута  $a_i$  базового отношения  $R$ :*

$\text{ndistinct}_A(R)$  – количество уникальных значений атрибута  $a_i$  в отношении  $R$ ;  
 $\text{min}_A(R)$ ,  $\text{max}_A(R)$  – минимальное и максимальное количество возможных значений атрибута  $a_i$  в отношении  $R$ ;

$\text{SC}_A(R)$  – кардинальность выборки атрибута  $a_i$  в отношении  $R$ . Оно представляет собой среднее количество кортежей, которые удовлетворяют логическим условиям  $L_{ik}$ , заданным для атрибута  $a_i$ .

Если предположить, что значения атрибута  $a_i$  равномерно распределены по отношению  $R$  и что существует по крайней мере одно значение, удовлетворяющее заданному условию, то значение  $\text{SC}_A(R)$  может быть определено из соотношений (5 – 6):

1 – если атрибут  $a_i$  является ключевым атрибутом отношения  $R$ ; (5)

$[\text{ntuples}(R)/\text{ndistinct}_A(R)]$  – в противном случае. (6)

Кроме того, может быть выполнена оценка кардинальности выборки для различных условий. В этом случае значение  $\text{SC}_A(R)$  определяется из соотношений (7 – 9):

$[\text{ntuples}(R) * ((\text{max}_A(R) - c) / (\text{max}_A(R) - \text{min}_A(R)))]$  – для  $(A > c)$ ; (7)

$[\text{ntuples}(R) * ((c - \text{max}_A(R)) / (\text{max}_A(R) - \text{min}_A(R)))]$  – для  $(A < c)$ ; (8)

$f(\text{ntuples}(R) / \text{ndistinct}_A(R)) * n$  – для  $(A \text{ in } \{C_1, c_2, \dots, c_n\})$ . (9)

*Для каждого многоуровневого индекса  $i$  множества значений атрибута  $a_i$ :*

$\text{nlevels}_A(i)$  – количество уровней индекса  $i$ ;

$\text{nlfblocks}_A(i)$  – количество листовых блоков индекса  $i$ .

Поскольку, используя (2 – 4), существует возможность сведения операций с кардинальностью вычисления операнда  $O(n^2)$  к операциям с кардинальностью вычисления операнда  $O(n)$ , их применение будет целесообразным при выполнении оценки базовых операций выборки, соединения и проекции.

**4. Последовательность получения оценки операторов ЯМД.** Рассмотрим последовательность получения оценки операторов ЯМД на примере операции выборки  $O(\text{select\_on}[P])$ .

Существует несколько различных способов реализации операции выборки, выбор которых зависит от структуры файла, в котором хранится соотношение, а также от того, какими являются используемые в

предикате атрибуты – индексированными или хешированными. Ниже перечислены основные типы стратегий:

- линейный поиск (файл не отсортирован, индекс отсутствует);
- двоичный поиск (упорядоченный файл, индекс отсутствует);
- поиск по равенству ключа перемешивания;
- поиск по равенству первичного ключа;
- поиск по неравенству первичного ключа;
- поиск по равенству значения кластерного (вторичного) индекса;
- поиск по равенству значения некластерного (вторичного) индекса;
- поиск по неравенству значения вторичного индекса типа В<sup>+</sup>-Tree.

Алгоритмы реализации указанных стратегий приведены в [2], их анализ позволил получить основные стоимостные характеристики рассматриваемой операции (табл. 2).

Таблица 2

Сводная таблица стоимости реализации операции выборки

<b>Стратегия</b>	<b>Оценка операции</b>
<i>линейный поиск</i>	[nblocks(P)/2] ; nblocks(P)
<i>двоичный поиск</i>	[log <sub>2</sub> (nblocks(P))]; [log <sub>2</sub> (nblocks(P))]+[SCA(P)/bfactor(P)]-1
<i>поиск по равенству первичного ключа</i>	nlevelsA(i)+1
<i>поиск по неравенству первичного ключа</i>	nlevelsA(i)+[nblocks(P)/2]
<i>поиск по равенству значения не- кластерного (вторичного) индекса</i>	nlevelsA(i)+[SCA(P)]
<i>поиск по неравенству значения вторичного индекса типа В<sup>+</sup>-Tree</i>	nlevelsA(i)+[nfblocksA(i)/2+ntuples(P)/2]

Таким образом, порядок получения оценки оператора ЯМД можно отобразить следующей последовательностью действий:

- определение класса оператора ЯМД;
- представление операторов ЯМД запроса абонентов эквивалентными операциями реляционной алгебры;
- определение кардинальности предиката оператора ЯМД;
- определение стратегии реализации оператора ЯМД в системе;
- модификация операций реляционной алгебры в зависимости от стратегии реализации, определенной в предыдущем пункте (в случае необходимости);
- определение глобальной оценки реализации оператора ЯМД.

**Выводы.** Подводя итог вышеизложенному, можно сказать, что процесс получения оценки операторов ЯМД является одним из этапов составления графика реализации запросов пользователей и транзакций. При этом, учитывая динамику изменения рабочей нагрузки в ТС, данный этап должен быть реализован в масштабе реального времени, поскольку актуальность информации в условно-постоянных системах не превышает 1 мин. Предложенный способ позволяет находить решение для алгоритмов, рассмотренных в [5, 6], за допустимое время в среднем для 75 элементарных операторов ЯМД, что покрывает область максимально возможной нагрузки СБД ТС. Полученная оценка может быть использована как системой, так и пользователем для составления графика реализации запросов пользователей и транзакций в СБД ТС.

### ЛИТЕРАТУРА

1. Томас Коннолли, Каролин Бег. Базы данных: проектирование, реализация и сопровождение. – М.: Вильямс, 2000. – 1114 с.
2. К. Дж. Дейт. Введение в системы баз данных, 6-е издание. – К.; М.; СПб.: Издательский дом «Вильямс», 1999. – 848 с.
3. Ачасова С.М., Бандман О.Л. Корректность параллельных вычислительных процессов. – Н-ск: Наука, 1990. – 253 с.
4. Кучук Г.А. Метод синтезу логічної структури мережевої бази даних // Системи обробки інформації. – Х. : НАНУ, ПАНМ, ХВУ, 2001. – Вип. 2(12). – С. 32-36.
5. Осиевский С.В. Способ решения задачи параллельной обработки запросов к базам данных // Системи обробки інформації. – Х: НАНУ, ПАНМ, ХВУ. – 2002. – Вип. 5(21). – С. 197 – 204.
6. Осиевский С.В., Тимченко А.А. Процедура определения точки входа на основе анализа векторов весовых характеристик исходного графа состояний // Системи обробки інформації. – Х: НАНУ, ПАНМ, ХВУ. – 2002. – Вип. 6(22). – С. 281 – 284.
7. Codd E.F.- A Relational Model of Data for Large Shared Data Banks // CACM. – 1970. – 13, JV's 6. (риєпНЗ,наНо: Milestones of Research – Selected Papers 1958-1982 (CACM 25th Anniversary Issue) // CACM. – 1983. – 26, № 1.)
8. Кучук Г.А. Формалізація предметної області багатовимірних баз даних // Системи обробки інформації. – Х. : ХВУ, 2001. – Вип. 1(11). – С. 110 - 114.
9. Кульба В.В., Ковалевский С.С., Косяченко С.А. Теоретические основы проектирования оптимальных структур распределенных баз данных. – М.: Синтез, 1999. – 461 с.

Поступила 1.02.2003

**Осиевский Сергей Валерьевич**, адъюнкт кафедры ХВУ. В 1997 году окончил Харьковский военный университет. Область научных интересов – управление процессами сопровождения и развития баз данных.