электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0421100025. ISSN 1994-0408

Оценка времени выполнения SQL-запросов к базам данных

77-30569/296143

 # 01, январь 2012

Григорьев Ю. А.

УДК 004.657

МГТУ им. Н.Э. Баумана
grigorev@iu5.bmstu.ru

Введение

В некоторых работах [1, 5, 6] для нахождения коэффициентов аналитических выражений, используемых для оценки времени выполнения запросов к базам данных, предлагается использовать калибровочную модель, представляющую собой определённую БД, набор запросов, а также аппаратно-программный комплекс, на котором выполняются калибрующие эксперименты. Предлагаемые аналитические выражения необходимо строить для каждой конфигурации аппаратно-программных средств. Более того, эти выражения не отражают особенностей выполнения сложных запросов SQL. Для проведения расчётов вводятся некоторые сложные интегрированные параметры, при этом не совсем понятна их природа и не указывается, как можно оценить эти параметры.

В работах [2, 7, 8] предлагаются методы оценки стоимостных характеристик различных способов соединения таблиц (NLJ, SMJ, HJ), используемых при выполнении запросов к базе данных. К сожалению, здесь не учитывается случайных характер параметров соединяемых таблиц. Т. е. расчёты ведутся на уровне средних значений. В этих работах также не показано, как можно получить исходные данные для расчётов характеристик соединения промежуточных таблиц, получаемых при реализации оптимального плана выполнения исходного запроса к базе данных.

В данной работе разрабатывается новый математический аппарат анализа времени выполнения запросов SELECT к распределённой базе данных, учитывающий механизмы декомпозиции сложных запросов на подзапросы и соединения результатов их выполнения, параметры логической схемы и случайную природу характеристик наполнения базы данных. Предлагаемый подход можно использовать на ранних этапах проектирования баз данных для оценки временных характеристик выполнения запросов различной степени сложности на различных аппаратно-программных платформах. При этом оценки получаются в виде преобразований Лапласа-Стилтьеса, что позволяет оценивать не только средние величины, но и дисперсии, а также моменты более высоких порядков.

1. Организация обработки запросов SQL

С точки зрения программистов распределённая база данных – это большая виртуальная БД. Оптимизатор запросов (планировщик запросов) СУБД автоматически выполняет декомпозицию запроса на подзапросы и организует их выполнение. При этом оптимизатор выполняет следующие шаги:

1. Исходный запрос преобразуется в формулу реляционной алгебры (явно или нет). Известно, что оператор SELECT языка SQL может быть представлен в виде следующей формулы реляционной алгебры [4]:

$$\pi_A (\sigma_F (R_1 \times R_2 \times ... \times R_n)), \qquad (1)$$

где $(R_1 \times R_2 \times ... \times R_n)$ – декартово произведение отношений (таблиц), указанных за ключевым словом FROM оператора SELECT,σ_F – операция селекции кортежей декартова произведения в соответствии с условием F, указанным за ключевым словом WHERE, π_A – проекции селекции на множество атрибутов A, которые перечисляются за ключевым словом SELECT.

2. Эта формула подвергается оптимизатором улучшению. Смысл улучшения заключается в перемещении операций селекции и проекции внутрь декартова произведения с использованием законов реляционной алгебры и некоторых правил, сформулированных в [4]. В результате формула (1) преобразуется к виду:

$$\pi_A(\sigma_{F^\wedge} (\underline{\pi_{A_1^\wedge} (\sigma_{F_1} (R_1))} \times \underline{\pi_{A_2^\wedge} (\sigma_{F_2} (R_2))} \times ...)) \qquad (2)$$

Здесь $F = F^\wedge \, and \; F_1 \; and \ldots and \; F_n, \;\; F -$исходное условие поиска в формуле (1), $F^\wedge -$условие соединения результатов выполнения подзапросов

$Q_i = \pi_{\hat{A_i}}(\sigma_{F_i}(R_i))$, F_i – условие поиска в i–ом подзапросе. Подчёркнутые в (2) элементы (подзапросы) имеют меньшую размерность, чем исходные отношения R_1 и R_2.

3. В зависимости от настроек, сделанных проектировщиком, оптимизатор использует один из следующих методов построения оптимального плана выполнения запроса на основе формулы, полученной на шаге 2: продукционная или стоимостная оптимизация.

4. При формировании оптимального плана на шаге 3 оптимизатор выбирает метод соединения таблиц. В основном используются следующие методы: соединение с помощью вложенных циклов (NLJ - Nested Loop Joins), соединение посредством сортировки-слияния (SMJ- Sort-Merge Join), хешированное соединение (HJ - Hash Joins), кластерное соединение.

Из выражений (1) и (2) следует, что описание запроса к базе данных и, следовательно, время его выполнения зависят от схемы базы данных. Из анализа работы оптимизаторов следует [3], что построение оптимального плана сводится, в основном, к определению последовательности выполнения соединений промежуточных таблиц, полученных при выполнении подзапросов:

$$Q = Q_{i_1} \times Q_{i_2} \times \cdots \times Q_{i_n} \quad (3)$$

здесь Q_{i_k} - промежуточная таблица, полученная при выполнении соответствующего подзапроса. Эти таблицы будем обозначать так же, как и подзапросы (см. (2)), Q - результирующая таблица, которая соответствует исходному запросу к базе данных. Из (3) следует, что оценка времени выполнения запроса сводится к получению оценок времени обработки подзапросов Q_k и построению рекуррентной математической процедуры оценки времени выполнения соединений. Введём некоторые обозначения.

1. Пусть A_i - множество атрибутов таблицы (отношения) R_i, которые входят в условие (предикат) F_i i-го подзапроса в (2), $i = \overline{1, n}$. В общем случае на атрибуты A_i в условии поиска F_i накладываются ограничения, которые могут изменяться при разных

обращениях к запросу. a_{ij} - атрибут из множества A_i, т. е. $A_i = \{a_{ij}\}_j$. По определению $\{a_{ij}\}_j = \{a_{i1}, a_{i2}, \ldots\}$.

2. $\{b_{ijk}\}_k$ - те значения атрибута a_{ij}, которые удовлетворяют соответствующему элементарному условию по a_{ij} в предикате F_i. Ясно, что $\{b_{ijk}\}_k \subseteq D_{ij} = \{d_{ijm}\}_m$, где D_{ij} - домен (множество значений) атрибута a_{ij} в таблице R_i. $|D_{ij}| = I_{ij}$ - мощность домена. Далее будем использовать следующее множество:

$$M_{ij} = \{m |\ d_{ijm} \subseteq \{b_{ijk}\}_k\}, \tag{4}$$

4. η_{ijm} - вероятность, что атрибут a_{ij} какого-нибудь кортежа (записи) таблицы R_i принимает значение d_{ijm}, $\sum_{m=1}^{|D_{ij}|} \eta_{ijm} = 1$. Вероятности η_{ijm} можно задать априори или получить из гистограмм, которые могут быть построены утилитой СУБД сбора статистик. Для записей из таблиц $\{R_i\}$ будем считать независимыми в совокупности события

$$\{a_{ij} = d_{ijm}\}_{ijm}. \tag{5}$$

5. Далее будем использовать преобразование Лапласа-Стилтьеса (Л.-С.) функции распределения случайного времени: $\Psi(s) = \int_0^{\infty} e^{-st} dF(t)$ (или просто преобразование Л.-С. времени). Для Л.-С. справедливо равенство:

$$\Psi^{(n)}(0) = (-1)^n M(\xi^n),$$

где $\Psi^{(n)}(0)$ - n-ая производная $\Psi(s)$ при $s = 0$, $M(\xi^n)$ - n-ый начальный момент случайной величины ξ. Например, $M(\xi)$ - это среднее значение, $M(\xi^2) - M^2(\xi)$ - дисперсия и т. д.

6. $G_i(z)$ - производящая функция числа записей в таблице R_i. Производящая функция дискретной случайной величины ξ с целыми неотрицательными значениями определяется следующим выражением:

$$\Gamma(z)=\sum_{i=0}^{\infty} p_i z^i\,,\quad \sum_{i=0}^{\infty} p_i = 1,$$

где p_i - вероятность, что случайная величина ξ равна i. Имеют место равенства

$$\Gamma^{(1)}(1)=M(\xi)\,,\quad \Gamma^{(2)}(1)=M(\xi^2)-M(\xi)\,,\quad \Gamma^{(n)}(0)=n!\,p_n\,,$$

где $\Gamma^{(n)}(z)$ - n-ая производная $\Gamma(z)$ в точке z.

2. Оценка времени обработки подзапросов

Этот раздел написан в соавторстве с Плутенко А.Д. Найдём преобразование Л.-С. времени чтения блоков индексов по тем атрибутам из A_i, для которых при выполнении подзапроса Q_i из (2) используются индексы.

Лемма 1. *Преобразование Л.-С. времени чтения блоков нижнего уровня по всем используемым в подзапросе Q_i индексам таблицы R_i равно:*

$$\tau_i(s)=\prod_{j\in N_i}\vartheta_{ij}(\delta_{ij}(s)),\tag{6}$$

здесь N_i - множество тех атрибутов из A_i, для которых при выполнении подзапроса Q_i используются индексы, $\delta_{ij}(s)$ - преобразование Л.-С. времени чтения одного блока нижнего уровня индекса по атрибуту a_{ij}, $\vartheta_{ij}(z)$ - производящая функция читаемых блоков индекса по атрибуту a_{ij}.

Если множество элементов $\{d_{ijm}\}_{m\in M_{ij}}$ домена D_{ij} после упорядочивания образуют последовательность смежных значений, то

$$\vartheta_{ij}(z)=\sum_k p_{ijk}z^k\,,\tag{7}$$

где $p_{ij0} = W_{ij}(0)$, $p_{ijk} = \sum_{l=(k-1)q_{ij}+1}^{kq_{ij}} \frac{W_{ij}^{(l)}(0)}{l!}$, $k \geq 1$,

$$W_{ij}(z) = G_i(1 - \sum_{m \in M_{ij}} \eta_{ijm}(1-z)),$$

q_{ij} - максимальное число записей в блоке нижнего уровня индекса по атрибуту a_{ij}.

В противном случае необходимо определить произведение выражений (7), которые соответствуют участкам смежных значений атрибутов в множестве $\{d_{ijm}\}_{m \in M_{ij}}$. Найдём теперь преобразование Л.-С. времени чтения блоков таблицы R_i с записями, удовлетворяющими условию поиска F_i. Но, прежде всего, определим рекуррентную процедуру расчёта вероятности χ_i, что произвольная запись таблицы R_i удовлетворяет условию поиска F_i. Вероятность, что запись таблицы R_i удовлетворяет элементарному условию поиска по атрибуту a_{ij} в предикате F_i, можно определить с помощью следующего выражения:

$$\chi_{ij} = \sum_{m \in M_{ij}} \eta_{ijm} \tag{8}$$

В частности из (8) следует, что если вероятности η_{ijm} одинаковы, то

$$\chi_{ij} = \frac{|M_{ij}|}{|D_{ij}|}, \tag{9}$$

где $|M_{ij}|$ - мощность множества M_{ij}(см. (4)), а $|D_{ij}|$ - мощность домена D_{ij}. Элементарные условия по атрибутам $\{a_{ij}\}_j$ в F_i могут быть связаны различными логическими условиями: AND, OR и может быть NOT. Для расчёта вероятности χ_i можно воспользоваться рекуррентной процедурой, которая описана в виде табл. 1.

Таблица 1

Условие	Вероятность
усл.3 = усл.1 AND усл.2	$p_{усл3} = p_{усл1} \cdot p_{усл2}$
усл.3 = усл.1 OR усл.2	$p_{усл3} = p_{усл1} + p_{усл2} - p_{усл1} \cdot p_{усл2}$
усл.2 = NOT(усл.1)	$p_{усл2} = 1 - p_{усл1}$

Здесь $p_{усл1}$, $p_{усл2}$, $p_{усл3}$ - это вероятности, что запись таблицы R_i удовлетворяет соответствующему условию. Первоначально "усл.1" и "усл.2" в правой части равенств первого столбца табл. 1 - это элементарные условия в F_i, а $p_{усл1}$ и $p_{усл2}$ - это вероятности, вычисляемые с помощью выражения (8).

Следует отметить, что χ_i не зависит от числа записей Nв таблице R_i, т. к. от N не зависят вероятности χ_{ij} (см. (8)). Применяя схему Бернулли с параметром χ_i и используя формулу полной вероятности, найдём производящую функцию числа записей таблицы R_i, удовлетворяющих условию поиска F_i:

$$V_i(z) = \sum_N p_{iN}(1 - \chi_i(1 - z))^N = G_i(1 - \chi_i(1 - z)), \tag{10}$$

здесь χ_i - вероятность, которая определяется с помощью описанной выше рекуррентной процедуры (см. табл. 1), $G_i(z)$ - производящая функция числа записей в таблице R_i.

<u>Лемма 2</u>. *Преобразование Л.-С. времени чтения блоков таблицы R_i с записями, удовлетворяющими условию поиска F_i, имеет вид*

$$t_i(s) = \gamma_i(\beta_i(s)), \tag{11}$$

где $\beta_i(s)$ - преобразование Л.-С. времени чтения одного блока таблицы R_i, $\gamma_i(z)$ - производящая функция читаемых блоков таблицы R_i.

Если записи, удовлетворяющие условию поиска F_i, располагаются в блоках таблицы R_i последовательно (в соседних строках), то

$$\gamma_i(z) = V_i(0) + \sum_{k=1}^{\infty} \Big(\sum_{l=(k-1)r_i+1}^{kr_i} \frac{V_i^{(l)}(0)}{l!} \Big) z^k, \tag{12}$$

здесь $V_i(z)$ определяется выражением (10), r_i- максимальное число записей в блоке таблицы R_i. В общем случае, если записи, удовлетворяющие условию поиска F_i, не располагаются в блоках таблицы R_i последовательно (в соседних строках), то производящую функцию читаемых блоков таблицы R_i можно найти с помощью произведения выражений вида (12), которые соответствуют участкам с последовательным размещением записей в блоках.

Теорема 1. *Для преобразования Л.-С. времени выполнения подзапроса Q_i справедливо следующее выражение:*

$$T_i(s) = \gamma_i(\beta_i(s)) \prod_{j \in N_i} \vartheta_{ij}(\delta_{ij}(s)), \tag{13}$$

где N_i - множество тех атрибутов из A_i, для которых при выполнении подзапроса Q_i используются индексы, $\delta_{ij}(s)$ - преобразование Л.-С. времени чтения одного блока нижнего уровня индекса по атрибуту a_{ij}, $\vartheta_{ij}(z)$ - производящая функция читаемых блоков индекса по атрибуту a_{ij}, которая определяется выражением (7), $\beta_i(s)$ - преобразование Л.-С. времени чтения одного блока таблицы R_i, $\gamma_i(z)$ - производящая функция читаемых блоков таблицы R_i, которая определяется выражением (12).

Утверждение теоремы 1 следует из формул (6) и (11), являющихся утверждениями лемм 1 и 2, и свойств преобразования Л.-С..

3. Рекуррентная формула для определения производящей функции числа кортежей соединяемых таблиц

Соединение промежуточных таблиц $\{Q_i\}$ выполняется попарно в последовательности, определённой при построении оптимального плана (см. (3)).

Соединение является рекуррентной математической процедурой: $Q := Q \triangleright\triangleleft Q_{i_k}$, $k = \overline{2,n}$. Первоначально $Q = Q_{i_1}$. Далее без потери общности будем считать, что $i_k = k$. Будем также полагать, что $A_i = \{a_{ij}\}_j$ - это множество всех атрибутов какой-нибудь соединяемой таблицы. Для оценки времени соединения таблиц (раздел 4) на каждом шаге рекуррентной процедуры необходимо выполнить следующие действия:

1. Найти производящую функцию числа записей и множество атрибутов для каждой из двух соединяемых таблиц.

2. Для каждого атрибута a_{ij} соединяемых таблиц определить его домен $D_{ij} = \{d_{ijm}\}_m$ и вероятности $\{\eta_{ijm}\}_m$.

Сначала покажем, как указанные выше задачи 1 и 2 решаются для соединяемых таблиц $\{Q_i\}$, которые соответствуют подзапросам.

1. Производящая функция для числа записей определяется выражением (10), а множество атрибутов совпадает с $\hat{A_i}$ (см. (2)).

2. Представим условие F_i в виде дизъюнктивной формы. Домен D_{ij} атрибута a_{ij} в таблице Q_i определим следующим образом:

а) если атрибут входит во все конъюнкты дизъюнктивной формы, то в этом случае полагаем

$$D_{ij} = \{d_{ijm}\}_m := \left\{ d_{ijm} \mid d_{ijm} \in \bigcup_e \{b_{ijk}^e\}_k \right\}, \qquad (14)$$

$\{b_{ijk}^e\}_k$ - это те значения атрибута a_{ij}, которые удовлетворяют элементарному условию по a_{ij} в конъюнкте с номером e,

б) иначе домен атрибута оставляем без изменения (как в таблице R_i).

Если домен атрибута изменился (см. (14)), то необходимо выполнить "нормирование" вероятностей элементов нового домена D_{ij}:

$$\eta_{ijm} := \frac{\eta_{ijm}}{\sum_{k=1}^{|D_{ij}|} \eta_{ijk}}, \quad m = \overline{1, |D_{ij}|} \qquad (15)$$

Покажем теперь, как

1) найти производящую функцию числа записей в промежуточной таблице $Q := Q \triangleright\triangleleft Q_k$, которая, в свою очередь, может соединяться с таблицей Q_{k+1}, а также множество атрибутов в таблице Q,

2) определить домены $D_{ij} = \{d_{ijm}\}_m$ атрибутов a_{ij}, добавленных в таблицу Q в результате соединения, и вероятности $\{\eta_{ijm}\}_m$.

Без потери общности будем считать, что соединяются таблицы Q_1 и Q_2, т. е. $Q := Q_1 \triangleright\triangleleft Q_2$, т. к. далее рекуррентно можно положить $Q_1 := Q$ и $Q_2 := Q_k$, $k = \overline{3, n}$. Рассмотрим общий случай соединения таблиц по нескольким атрибутам. Без потери общности будем считать, что таблицы Q_1 и Q_2 соединяются соответственно по атрибутам $\{a_{11}, \ldots, a_{1r}\}$ и $\{a_{21}, \ldots, a_{2s}\}$. Пусть φ - условие связи между этими группами атрибутов.

В дальнейшем под *соединяемыми записями* таблиц Q_1 и Q_2 будем понимать записи этих таблиц, в которых значения атрибутов связи { $a_{11} = d_{11m_1} \in D_{11}$,..., $a_{1r} = d_{1rm_r} \in D_{1r}$} и { $a_{21} = d_{21l_1} \in D_{21}$,..., $a_{2s} = d_{2sl_s} \in D_{2s}$} связаны отношением φ, т. е. $\{d_{11m_1}, \ldots, d_{1rm_r}\} \varphi \{d_{21l_1}, \ldots, d_{2sl_s}\}$, где φ - условие связи таблиц Q_1 и Q_2 по атрибутам связи $\{a_{11}, \ldots, a_{1r}\}$ и $\{a_{21}, \ldots, a_{2s}\}$ (=, <, ≤, >, ≥, ≠ или более сложное условие). Под *группой соединяемых записей* будем понимать совокупность соединяемых записей таблицы Q_1 (или Q_2), в которых атрибуты связи $\{a_{11}, \ldots, a_{1r}\}$ (или $\{a_{21}, \ldots, a_{2s}\}$) принимают одинаковые значения, т. е. имеют одинаковые подстроки в соответствующей таблице.

Лемма 4. *Пусть количества кортежей в разных группах соединяемых записей таблиц Q_1 и Q_2 независимы в совокупности. Тогда справедливы следующие утверждения:*

1. Производящая функция числа записей в таблице $Q := Q_1 \triangleright\triangleleft Q_2$ имеет следующий вид:

$$V_Q(z)=\sum_N p_{2N}V_1\Big(\prod_{m_1=1}^{|D_{11}|}\dots\prod_{m_r=1}^{|D_{1r}|}(1-\prod_{j=1}^{r}\eta_{1jm_j}(1-$$
$$-(1-\sum_{(l_1,\dots,l_s)\in L_{m_1\dots m_r}}\prod_{j=1}^{s}\eta_{2jl_j}(1-z))^N))\Big), \tag{16}$$

где $p_{2N}=\dfrac{V_2^{(N)}(0)}{N!}$ *- вероятность, что число записей в таблице* Q_2 *равно* N *(см. свойства производящей функции в разделе 1),*

$V_2(z)$ *- производящая функция числа записей таблицы* Q_2, $V_1(z)$ *- производящая функция числа записей таблицы* Q_1,

$L_{m_1\dots m_r}=(l_1,\dots,l_s\,|\,\{d_{11m_1},\dots,d_{1rm_r}\}\,\varphi\,\{d_{21l_1},\dots,d_{2sl_s}\})$, φ *- это отношение связи таблиц* Q_1 *и* Q_2 *по атрибутам связи* $\{a_{11},\dots,a_{1r}\}$ *и* $\{a_{21},\dots,a_{2s}\}$ *(=, <,* $\le$*, >,* $\ge$*,* $\ne$ *или более сложное условие).*

Множество атрибутов таблицы Q *равно* $A_Q=A_1\cup A_2$*, где* A_1 *и* A_2 *- множества атрибутов таблиц* Q_1 *и* Q_2.

2. Для доменов атрибутов связи $\{a_{11},\dots,a_{1r}\}$ *и* $\{a_{21},\dots,a_{2s}\}$ *, вошедших в* Q*, справедливы следующие выражения:*

$$D_{1j}=(d_{1jm_j}\,|\,\exists m_1,\dots,m_{j-1},m_{j+1},\dots,m_r,l_1,\dots,l_s(\{d_{11m_1},\dots,d_{1jm_j},\dots,d_{1rm_r}\}\,\varphi\,\{d_{21l_1},\dots,d_{2sl_s}\})),\ j=\overline{1,r}, \tag{17}$$

$$D_{2j}=(d_{2jl_j}\,|\,\exists m_1,\dots,m_r,l_1,\dots,l_{j-1},l_{j+1},\dots,l_s(\{d_{11m_1},\dots,d_{1rm_r}\}\,\varphi\,\{d_{21l_1},\dots,d_{2jl_j},\dots,d_{2sl_s}\})),\ j=\overline{1,s}.$$

Вероятности появления элементов доменов D_{1j} *и* D_{2j} *в кортежах таблицы* Q *равны*

$$\eta_{1jm}:=\frac{\eta_{1jm}}{\sum_{k=1}^{|D_{1j}|}\eta_{1jk}},\ m=\overline{1,|D_{1j}|},\ j=\overline{1,r}, \tag{18}$$

$$\eta_{2jl} := \frac{\eta_{2jl}}{\sum_{k=1}^{|D_{2j}|} \eta_{2jk}}, \ l = \overline{1, |D_{2j}|}, \ j = \overline{1, s}.$$

Домены остальных атрибутов таблицы Q и вероятности появления элементов этих доменов в кортежах Q не изменяются.

4. Оценка времени выполнения соединения таблиц

В этом разделе выводятся преобразования Л.-С. времени соединения $Q = Q_1 \rhd\lhd Q_2$, которое может быть выполнено оптимизатором с помощью одного из методов, указанных в разделе 1. Производящие функции $V_1(z)$ и $V_2(z)$ числа записей в соединяемых таблицах Q_1 и Q_2 были получены в разделах 2 и 3 (см. формулы (10) и (16)). Производящая функция числа записей $V_Q(z)$ в таблице $Q = Q_1 \rhd\lhd Q_2$ была получена в разделе 3 (см. рекуррентную формулу (16)).

4.1. Метод соединения с помощью вложенных циклов NLJ

<u>Теорема 2</u>. *Преобразование Л.-С. времени ξ_U выполнения соединения таблиц Q_1 и Q_2 методом NLJ имеет вид:*

$$\Psi_{\xi_U}(s) = V_1(V_2(\Psi_C(s))) \cdot V_Q(\Psi_J(s)), \qquad (19)$$

где $V_1(z)$, $V_2(z)$, $V_Q(z)$ - производящие функции числа записей в таблицах Q_1, Q_2 и $Q = Q_1 \rhd\lhd Q_2$, $\Psi_C(s)$ - преобразование Л.-С. времени сравнения атрибутов связи двух кортежей из Q_1 и Q_2, $\Psi_J(s)$ - преобразование Л.-С. времени соединения двух кортежей из Q_1 и Q_2.

4.2. Метод соединение посредством сортировки-слияния SMJ

Сначала таблицы Q_1 и Q_2 при необходимости сортируются по атрибутам связи, а затем выполняется их соединение. Предположим, что сортировка выполняется в порядке возрастания атрибута связи и элементы домена этого атрибута также упорядочены в порядке возрастания: $d_{i11} < d_{i12} < \ldots < d_{i1I_{i1}}$, здесь I_{i1} - мощность домена D_{i1}, $i=1,2$. Также предположим, что сортировка выполняется обычным способом: путём перемещения влево k-ой записи внутрь уже упорядоченных $k-1$ записей (пузырьковый способ).

Лемма 5. *Пусть число перемещений записей на k-ом шаге не зависит от числа перемещений на предыдущих шагах. Тогда производящая функция числа перемещений при сортировке записей таблицы Q_i по атрибуту связи a_{i1} имеет вид*

$$H_{i1}(z) = \sum_{N} p_{iN} \prod_{k=1}^{N} \Big(\sum_{m=1}^{I_{i1}} \big(\eta_{i1m}(1 - \sum_{l=m+1}^{I_{i1}} \eta_{i1l}(1-z))^{k-1}\big)\Big), \qquad (20)$$

здесь p_{iN} - вероятность, что количество записей в таблице Q_i равно N.

Теперь найдём производящую функцию числа сравнений атрибутов связи соединяемых таблиц Q_1 и Q_2.

Лемма 6. *Производящая функция количества сравнений атрибутов связи при соединении таблиц Q_1 и Q_2 имеет следующий вид:*

$$E(z) = V_1(1 - \sum_{m=1}^{\hat{I}_{11}} \eta_{11m}(1-z)) \cdot V_2(1 - \sum_{m=1}^{\hat{I}_{21}} \eta_{21m}(1-z)), \qquad (21)$$

здесь

$$\hat{I}_{i1} = \max_{m \in M_i} m,\ M_i = \{m |\ d_{i1m} \le d\},\ i=1,2,\ d = \min_{i=1,2} \max_{m} d_{i1m},$$

$V_i(z)$ *- производящая функция числа записей в таблице* Q_i, $i=1,2$ *(см. формулы (10) и (16)).*

Теорема 3. *Преобразование Л.-С. времени ξ_U выполнения соединения таблиц Q_1 и Q_2 методом SMJ имеет вид:*

$$\Psi_{\xi_U}(s) = [H_{11}(\Psi_T(s))] \cdot [H_{21}(\Psi_T(s))] \cdot E(\Psi_C(s)) \cdot V_Q(\Psi_J(s)), \qquad (22)$$

$H_{i1}(z)$ *- производящая функция числа перемещений при сортировке записей таблицы Q_i по атрибуту связи a_{i1}, которая определяется выражением (20),* $i=1,2$,

$E(z)$ *- производящая функция количества сравнений атрибутов связи при соединении* *таблиц* Q_1 *и* Q_2, *которая определяется выражением (21),* $V_Q(z)$ *- производящая функция числа записей в таблице* $Q = Q_1 \rhd\lhd Q_2$ *(см. формулу (16)),* $\Psi_T(s)$ *- преобразование Л.-С. времени перемещения двух кортежей при сортировке таблиц* Q_1 *и* Q_2, $\Psi_C(s)$ *- преобразование Л.-С. времени сравнения атрибутов связи двух кортежей из* Q_1 *и* Q_2, $\Psi_J(s)$ *- преобразование Л.-С. времени соединения двух кортежей из* Q_1 *и* Q_2.

4.3. Метод хешированного соединения HJ

В процессе хешированного соединения выполняются следующие шаги:

1. Выбирается хеш-функция.

2. Записи соединяемых таблиц Q_1 и Q_2 хешируются по атрибутам связи, т. е. создаются две группы таблиц $\{Q_{1j}\}$, $\{Q_{2j}\}$, число таблиц в группе равно количеству разделов.

3. Для каждого раздела с помощью метода SMJ или NLJ выполняется соединение таблиц соответствующих таблиц: $\Omega_j = Q_{1j} \rhd\lhd Q_{2j}$, $j = \overline{1, r}$, r - число разделов.

4. Объединяются результаты соединений, выполненных на шаге 3: $Q = \bigcup_{j=1}^{r} \Omega_j$.

В качестве хеш-функции можно выбрать, например, деление по модулю 10 значений атрибутов связи. В этом случае число разделов $r = 10$ (число различных остатков от деления на 10) и в таблицу Q_{ij} попадут записи таблицы Q_i, для которых остаток от деления на 10 значения атрибута связи равен j ($j = \overline{0,9}$).

<u>Лемма 7.</u> *Справедливы следующие утверждения:*

1. Производящая функция числа записей в таблице Q_{ij}*(* $j = \overline{1, r}$, $i = 1,2$*) имеет вид*

$$V_{ij}(z) = V_i\Big(1 - \sum_{m \in K_i} \eta_{i1m}(1 - z)\Big), \tag{23}$$

здесь $V_i(z)$ *- производящая функция числа записей в таблице* Q_i, $K_i = \{m | h(d_{i1m}) = j\}$, h *- хеш-функция, которая возвращает номер раздела,* d_{i1m} *- элемент домена* D_{i1} *атрибута связи* a_{i1} *таблицы* Q_i.

2. Домен атрибута связи a_{ij1} *таблицы* Q_{ij} *включает следующие элементы:*

$$D_{ij1} = \{d_{i1m} | h(d_{i1m}) = j\} = \{d_{ij1m}\}_m. \tag{24}$$

Вероятности появления элементов доменов D_{ij1} *в кортежах таблицы* Q_{ij} *равны*

$$\eta_{ij1m} := \frac{\eta_{ij1m}}{\sum_{k=1}^{|D_{ij1}|} \eta_{ij1k}}, \ m = \overline{1, |D_{ij1}|}, \ j = \overline{1, r}, \ i = \overline{1, 2}, \tag{25}$$

здесь в правой части $\eta_{ij1m} = \eta_{i1k} : h(d_{i1k}) = j$.

Домены остальных атрибутов таблицы Q_{ij}, *вошедших в* Q_{ij} *из* Q_i, *и вероятности появления элементов этих доменов в кортежах* Q_{ij} *не изменяются.*

<u>Теорема 4</u>*. Преобразование Л.-С. времени* ξ_U *выполнения соединения таблиц* Q_1 *и* Q_2 *методом HJ имеет вид:*

$$\Psi_{\xi_U}(s) = \prod_{i=1}^{r} \Psi_{\xi_U i}(s), \tag{26}$$

где r *- число разделов,* $\Psi_{\xi_U i}(s)$ *- это преобразование Л.-С. времени соединения таблиц* Q_{1j} *и* Q_{2j}, *выполненного методом NLJ или SMJ. Формулы для* $\Psi_{\xi_U i}(s)$ *определяются выражениями (19) или (22). Производящие функции числа записей соединяемых таблиц* $Q_1 := Q_{1j}$ *и* $Q_2 := Q_{2j}$, *а также домены атрибутов связи определяются выражениями (23) и (24),* $j = \overline{1, r}$.

Доказательство теоремы 4 следует из описания алгоритма хешированного соединения, леммы 7 и свойств преобразования Л.-С.

5. Оценка времени выполнения исходного запроса SQL

На основании результатов, сформулированных в предыдущих разделах, можно получить преобразование Л.-С. времени выполнения исходного запроса (3).

Теорема 5. *Преобразование Л.-С. времени выполнения исходного запроса (3) имеет следующий вид:*

$$\Psi(s) = \prod_{i=1}^{n} T_i(s) \prod_{j=1}^{n-1} \Psi_{j\xi_U}(s), \tag{27}$$

где $T_i(s)$ *- это преобразование Л.-С. времени выполнения подзапроса* Q_i*, которое определяется выражением (13),* $\Psi_{i\xi_U}(s)$ *- преобразование Л.-С. времени* j*-го соединения промежуточных таблиц, которое определяет одним из выражений (19), (22) и (26) (тип соединения назначается оптимизатором для каждого* j*-го соединения),* n *- число подзапросов в (3).*

Доказательство теоремы 5 следует из свойств преобразования Л.-С. и теорем 1, 2, 3, 4, приведённых в предыдущих разделах. Используя выражение (27), можно оценивать различные моменты (среднее, дисперсию и др.) времени выполнения запросов SQL.

Важно подчеркнуть, что полученные выражения для преобразований Л.-С. времени выполнения подзапросов (13) и соединений (19), (22), (26), а также для производящих функций числа кортежей в исходных и результирующих таблицах (см. формулы (10) и (16)) могут быть использованы не только для расчёта характеристик времени выполнения запросов SQL (27), но и для оценки параметров функций распределений при подготовке исходных данных моделей массового обслуживания, которые часто используются для анализа показателей качества распределённых систем обработки данных на макроуровне.

Предложенный математический аппарат был использован при разработке Комплекса инструментальных Средств Анализа Моделей доступа к базам данных распределённых систем обработки данных (КСАМ). Этот комплекс относится к классу экспертных систем и предназначен для проведения вычислительных экспериментов с целью анализа временных показателей систем, основу которых составляют распределённые базы данных и приложения.

СПИСОК ЛИТЕРАТУРЫ

1. *Weimin Du, Ravi Krishnamurthy, Ming-Chien Shan*. Query Optimization in a Heterogeneous DBMS // Proceedings of 18th International Conference on Very Large Data Bases, August 23-27, 1992, Vancouver, Canada. - P. 277-291.

2. *Goetz Graefe*. Query Evaluation Techniques for Large Databases // ACM Computing Surveys. - 1993. - Vol. 25, № 2. - P. 73-170.

3. *Чаудхари С.* Методы оптимизации запросов в реляционных системах // Системы управления базами данных. - М., 1998. - № 3. - С. 22-36.

4. *Ульман Дж.* Основы систем баз данных. – М.: Финансы и статистика, 1983. – 334 с.

5. *Georges Gardarin, Jean-Robert Gruser, Zhao-Hui Tang*. Cost-based Selection of Path Expression Processing Algorithms in Object-Oriented Databases // Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India. - P. 390-401.

6. *G. Gardarin, F. Sha, and Z.-H. Tang.* Calibrating the query optimizer cost model of IRO-DB, an objectoriented federated database system // Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India. - P. 378--389.

7. *Mishra P., Eich M.H.* Join Processing in relational databases. // ACM Computing Surveys. - 1992. - Vol. 24, № 1.

8. *E. P. Harris, K. Ramamohanarao*. Join algorithm costs revisited // The VLDB Journal. - 1996. - Vol. 5, № 1. - P. 64--84.

electronic scientific and technical periodical
SCIENCE and EDUCATION
EL № FS 77 - 30569. №0421100025. ISSN 1994-0408

Estimation of SQL-query execution time

77-30569/296143

01, January 2012

Grigor'ev Yu.A.

Bauman Moscow State Technical University
grigorev@iu5.bmstu.ru

Mathematical apparatus of analysis of SELECT-query execution time in distributed database, considering the mechanisms of complex query decomposition into sub-queries and connection of their results, the parameters of logic scheme and random characteristics of database filling. Proposed approach could be used at early stages of design for estimation of time characteristics of execution of different complex queries on different hardware platforms. In this manner, estimation was in form of Laplase-Stieltjes transformation, that allowed to estimate not only average values but also dispersions and other issues.

Publications with keywords: optimization of SELECT operator, SQL-query, query processing time, generating function, data base, Laplase-Stieltjes transformation
Publications with words: optimization of SELECT operator, SQL-query, query processing time, generating function, data base, Laplase-Stieltjes transformation

Reference

1. Weimin Du, Ravi Krishnamurthy, Ming-Chien Shan, Query Optimization in a Heterogeneous DBMS, in: Proceedings of 18th International Conference on Very Large Data Bases, August 23-27, 1992, Vancouver, Canada, pp. 277-291.
2. Goetz Graefe, Query Evaluation Techniques for Large Databases, ACM Computing Surveys 25 (2) (1993) 73-170.
3. Chaudkhari S., Methods of query optimization in relational systems, Sistemy upravleniia bazami dannykh 3 (1998) 22-36.
4. Ul'man Dzh., Fundamentals of database systems, Moscow, Finansy i statistika, 1983, 334 p.
5. Georges Gardarin, Jean-Robert Gruser, Zhao-Hui Tang, Cost-based Selection of Path Expression Processing Algorithms in Object-Oriented Databases, in: Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India, pp. 390-401.

6. G. Gardarin, F. Sha, Z.-H. Tang, Calibrating the query optimizer cost model of IRO-DB, an objectoriented federated database system, in: Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India, pp. 378-389.

7. Mishra P., Eich M.H., Join Processing in relational databases, ACM Computing Surveys 24 (1) (1992) 63-113.

8. E. P. Harris, K. Ramamohanarao, Join algorithm costs revisited, The VLDB Journal 5 (1) (1996) 64-84.