

ОЦЕНКА ВРЕМЕНИ ВЫПОЛНЕНИЯ ЗАПРОСОВ С КОРРЕЛИРОВАННЫМИ ПОДЗАПРОСАМИ И ОПЕРАЦИЯМИ АГРЕГИРОВАНИЯ

Предложен подход к построению альтернативных планов выполнения запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования. Дана оценка временным характеристикам этих планов и рекомендации по выбору того или иного плана в зависимости от параметров запросов и наполнения базы данных.

В настоящее время в различных информационных системах (ИС) используются системы управления базами данных (СУБД), поддерживающие реляционную модель данных. Для описания и манипулирования данными применяется стандартный язык *SQL*, который относится к классу непроцедурных языков. *SQL*-запрос, поступивший на сервер базы, подвергается оптимизации в целях уменьшения времени его выполнения. Суть оптимизации состоит в генерации нескольких альтернативных планов реализации запроса и выборе плана с наименьшей стоимостью, пропорциональной времени выполнения запроса. В теории реляционных баз данных наибольшее внимание уделяется построению оптимального плана для класса запросов, включающих операции соединения таблиц или имеющих конструкции, которые можно свести к соединению [1, 2]. В теории по существу не рассматривается важный класс запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования. В некоторых публикациях [3, 4] в основном анализируются запросы с двумя коррелированными подзапросами (один из них внешний) и необоснованно утверждается, что для них оптимальным является план с группированием значений атрибута связи. При этом расчет стоимости и сравнение альтернативных планов не выполняются. Это приводит к тому, что во многих СУБД (*Oracle*, *DB2* и др.) для реализации запросов из указанного класса используются планы, основанные на операции группирования. В результате анализа выявили, что эти планы являются оптимальными только в некоторой области изменения параметров запросов и наполнения базы данных, т.е. часто для реализации запроса в СУБД выбирается неоптимальный план. Поэтому важной является задача генерации альтернативных планов выполнения запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования, оценки стоимости планов и построения областей их предпочтительного использования.

Преобразование исходного плана выполнения запроса в альтернативный план. На рис. 1 приведено описание исследуемого класса запросов к базе данных. Здесь используются следующие обозначения: R_i ($i = \overline{1, n}$) — таблица i -го подзапроса, из которой читаются записи в процессе выполнения запроса. Некоторые или все таблицы могут совпадать; A_{10} — атрибут таблицы R_1 , который является результатом выполнения запроса (A_{10} может быть подмножеством атрибутов или операций агрегирования); A_{i1} ($i = \overline{1, n-1}$) — атрибут i -й таблицы, для которого установлено подусловие поиска $A_{i1} \tau_i f_{i+1}(A_{i+1}, 2)$, τ_i ($i = \overline{1, n-1}$) — арифметический оператор сравнения ($=, >, >=, <$ и т.д.), f_i ($i = \overline{2, n}$) — одна из операций агрегирования ($MIN, MAX, SUM, COUNT, AVG$ и др.). Это может быть и арифметическое выражение, куда входят несколько операций агрегирования; A_{i2} ($i = \overline{2, n}$) — атрибут i -й таблицы, являющийся аргументом функции агрегирования f_i . Агрегирование значений A_{i2} выполняется для тех записей таблицы R_i , которые удовлетворяют условию $WHERE$ i -го вложенного подзапроса. Будем считать, что 1-й вложенный подзапрос — это самый внешний запрос $SELECT$; A_{i3} ($i = \overline{1, n-1}$) — атрибут i -й таблицы, по которому устанавливается соединение с $(i+1)$ -й таблицей (R_{i+1}) в $(i+1)$ -м подзапросе (поэтому подзапрос является коррелированным); A_{i4} ($i = \overline{2, n}$) — атрибут i -й таблицы, по которому устанавливается соединение с $(i-1)$ -й таблицей (R_{i-1}) в i -м подзапросе.

На рис. 2, *a* показан результат частичной трансляции описания запроса (см. рис. 1) в дерево выражений реляционной алгебры (π и σ — операции проекции и селекции). На каждом i -м уровне вложенности ($i = \overline{1, n-1}$) введена операция σ без индекса, т.е. без условия,

```

SELECT A10      FROM R1
WHERE A11  $\tau_1$ 
      (SELECT f2(A22)      FROM R2
      WHERE R1.A13=R2.A24 AND A21 2
            (SELECT      f3(A32)      FROM R3
            WHERE      R2.A23=R3.A34 AND A31  $\tau_3$ 
                  (SELECT      f4(A42)      FROM R4
                  ...
                                                    AND An-1,1  $\tau_{n-1}$ 
                    (SELECT      fn(An2) FROM Rn
                    WHERE      Rn-1.An-1,3=Rn.An4
                    ) ...)))

```

Рис. 1. Описание исследуемого класса запросов

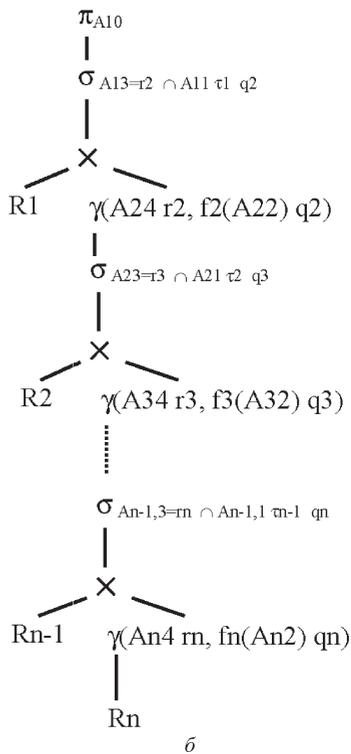
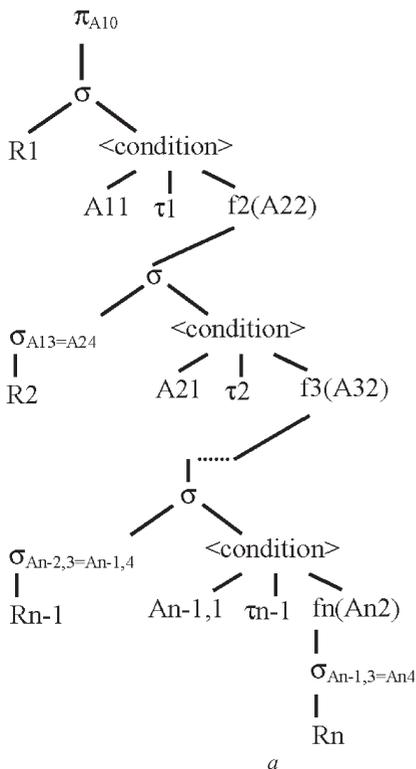


Рис. 2. Исходный (а) и альтернативный (б) планы выполнения запроса

которая определяет отношение, получаемое после выполнения операции селекции с условием $A_{i1} \tau_i f_{i+1}(A_{i+1,2})$ над отношением R_i или $\sigma_{A_{i-1,3} = A_{i4}}$. В процессе физической реализации дерево просматривается сверху вниз. Сначала читаются записи из R_1 . Для каждой такой записи определяется отношение $\sigma_{A_{13} = A_{24}}$ (значение A_{13} уже определено), далее для каждой записи из этого отношения определяется $\sigma_{A_{23} = A_{34}}$ (значение A_{23} уже определено) и так далее до самого нижнего уровня вложенности. Затем для каждой записи из рассматриваемых отношений $\sigma_{A_{i-1,3} = A_{i4}}$ выполняется перемещение снизу вверх, связанное с проверкой условий σ (без индекса) и вычислением функций агрегирования f_i .

Для построения альтернативного плана необходимо выполнять следующие шаги на каждом уровне вложенности.

1. Селекцию $\sigma_{A_{i-1,3} = A_{i4}}$ переместить вверх по дереву за операцию селекции σ без индекса, т.е. после ее выполнения.

2. Операцию селекции $\sigma_{A_{i3} = A_{i+1,4}}$ переместить за операцию селекции σ без индекса. На место перемещенной селекции вставить операцию группирования $\gamma(A_{i+1,4} r_{i+1}, f_{i+1}(A_{i+1,2}) q_{i+1})$. Здесь $A_{i+1,4}$ — это атрибут, по которому выполняется группирование. Группируемое

отношение поступает на вход операции γ после выполнения операций, расположенных по дереву ниже. $f_{i+1}(A_{i+1}, 2)$ — это операция агрегирования атрибута $A_{i+1}, 2$, которая выполняется для каждой группы. Идентификаторы g_{i+1} и q_{i+1} — это псевдонимы значений, указанных слева от них, т.е. $A_{i+1}, 4$ и $f_{i+1}(A_{i+1}, 2)$ соответственно.

3. Операцию селекции σ без индекса и условие «condition» заменить на селекцию $\sigma_{A_{i1} \tau_i q_{i+1}}$ над декартовым произведением R_i и γ .

4. Каскад селекций $\sigma_{A_{i3} = A_{i+1}, 4}$ и $\sigma_{A_{i1} \tau_i q_{i+1}}$ заменить на одиночную селекцию $\sigma_{A_{i3} = g_{i+1} \cap A_{i1} \tau_i q_{i+1}}$.

Последовательно применяя процедуру 1–4 для каждого уровня вложенности (i меняется от $(n - 1)$ до 1, селекция $\sigma_{A_{03} = A_{14}}$ считается пустой), план (см. рис. 2, а, исходный план) можно преобразовать к дереву, приведенному на рис. 2, б (альтернативный план). Используя законы реляционной алгебры, можно доказать, что преобразования 1–4 являются корректными, т.е. после выполнения снизу вверх процесса, представленного на рис. 2, б, будет получен правильный результат поиска данных.

В некоторых СУБД, например *Oracle*, план (см. рис. 2, б) выполняется неэффективно на физическом уровне (здесь соединение $\sigma_{A_{i3} = g_{i+1} \cap A_{i1} \tau_i q_{i+1}}$ реализуется путем фильтрации всех записей уровня i). Далее приведены n *SQL*-операторов (рис. 3), совокупность которых соответствует плану на рис. 2, б. В каждом запросе выполняется соединение двух таблиц, кроме самого первого. Практически во всех СУБД предусмотрены эффективные методы соединения. Поэтому эти n *SQL*-операторов часто выполняются быстрее, чем один исходный запрос (см. рис. 1) в соответствии с планом на рис. 2, б.

Оценка времени выполнения запроса для исходного и альтернативного планов. В настоящей работе доказаны приведенные ниже четыре теоремы о производящих функциях (ПФ) числа обработанных записей и преобразованиях Лапласа–Стилтьеса (ПЛС) времени выполнения запроса для исходного и альтернативного планов. Это важно, так как на этапе проектирования ИС конкретные значения полей записей базы данных и некоторые параметры запросов неизвестны, и поэтому можно говорить об их случайном характере.

Исходный план.

Теорема 1. Производящая функция $H_i(z)$ числа записей, обработанных на i -м уровне вложенности и ниже, для каждой записи ($i - 1$)-го уровня определяется следующей рекуррентной формулой:

$$H_i(z) = G_i(1 - p_i(1 - zH_{i+1}(z))), \quad i = \overline{1, n}, \quad (1)$$

где

$$G_i(z) = z^{V_i} \quad (2)$$

производящая функция числа записей в таблице R_i , V_i — прогнозируемое число записей в этой таблице,

$$p_i = \begin{cases} 1, & \text{для } i = 1; \\ \frac{1}{I_{i4}}, & \text{для } i = \overline{2, n}, \end{cases}, \quad (3)$$

I_{i4} — прогнозируемая мощность атрибута A_{i4} (число разных значений), $H_{n+1}(z) \equiv 1$.

Доказательство. Из описания процесса реализации плана (см. рис. 2, а) следует, что число записей, обработанных на i -м уровне вложенности, определяется селекцией $\sigma_{A_{i4}=a}$, где a — значение атрибута $A_{i-1,3}$ записи $(i-1)$ -го уровня ($i = \overline{2, n}$). Для $i = 1$ селекция отсутствует. Из работы [2] (см. формулу (2.25)) следует, что вероятность записи таблицы, удовлетворяющей условию $\sigma_{A_{i4}=a}$, равна $p_i = \frac{1}{I_{i4}}$, $i = \overline{2, n}$; для $i = 1$ $p_1 = 1$, так как селекция отсутствует.

Используя формулу (2.20) из работы [2], получим, что ПФ $H_i^*(z)$ числа записей, обработанных только на i -м уровне вложенности для каждой записи $(i-1)$ -го уровня равна

$$H_i^*(z) = G_i(1 - p_i(1 - z)). \quad (4)$$

Используя дополнительное событие [5], ПФ $H_i^*(z)$ можно трактовать как вероятность, что записи, обработанные только на i -м уровне вложенности для каждой записи $(i-1)$ -го уровня, являются “красными” (вероятность этого дополнительного события равна z). Выражение $zH_{i+1}(z)$ можно интерпретировать как вероятность того, что запись i -го уровня вложенности является “красной” (z) и все связанные с ней записи более нижнего уровня также являются “красными” ($H_{i+1}(z)$). Подставляя $zH_{i+1}(z)$ в выражение (4) вместо z , получим вероятность, что записи, обработанные на i -м уровне вложенности и ниже, для каждой записи $(i-1)$ -го уровня, являются “красными”. А эта вероятность и есть $H_i(z)$, т.е. $H_i(z) = H_i^*(zH_{i+1}(z))$.

Выражение (1) доказано.

Следствие 1. Производящая функция числа записей базы данных, обработанных при выполнении запроса в соответствии с исходным планом (см. рис. 2, а), равна

$$H(z) = H_1(z). \quad (5)$$

Это следует из определения ПФ $H_1(z)$ (см. (1)).

Теорема 2. ПЛС $T_i(s)$ времени обработки записей i -го уровня вложенности и ниже для каждой записи ($i - 1$)-го уровня равно

$$T_i(s) = G_i(1 - p_i(1 - \delta_i(s)T_{i+1}(s)\xi_i(s))), \quad i = \overline{1, n}, \quad (6)$$

где $G_i(z)$ и p_i определяются выражениями (2) и (3); $\delta_i(s)$ — ПЛС-времени поиска/чтения одной записи i -го уровня вложенности из БД; $\xi_i(s)$ — ПЛС-времени сравнения в соответствии с условием $Ai1 \tau i \text{ fi}+1(Ai+1,2)$ для $i = \overline{1, n-1}$ и/или определения текущего значения функции агрегирования $\text{fi}(Ai2)$ при успешном сравнении для $i = \overline{2, n}$ (см. рис. 2, а),

$$T_{n+1}(s) \equiv 1.$$

Доказательство. При доказательстве *теоремы 1* была получена производящая функция $H_i^*(z)$ числа записей, обработанных только на i -м уровне вложенности для каждой записи ($i - 1$)-го уровня (см. уравнение (4)). ПЛС-времени обработки такой записи и связанных с ней записей нижних уровней определяется следующим выражением:

$$T_i^{\sim}(s) = \delta_i(s)T_{i+1}(s)\xi_i(s). \quad (7)$$

Для доказательства (7) воспользуемся методом дополнительного события [5], трактуя ПЛС как вероятность того, что за случайное время не наступит “катастрофы” (s — интенсивность “катастроф”). Тогда вероятность $T_i^{\sim}(s)$ равна вероятности того, что “катастрофа” не наступит за время:

- 1) поиска/чтения записи i -го уровня ($\delta_i(s)$);
- 2) обработки связанных с ней записей нижних уровней ($T_{i+1}(s)$);
- 3) сравнения атрибута $Ai1$ и агрегирования значения атрибута $Ai2$ ($\xi_i(s)$).

Это доказывает выражение (7). Подставляя (7) в (4) и учитывая свойства ПФ и ПЛС, получим утверждение теоремы, т.е. $T_i(s) = H_i^*(\delta_i(s)T_{i+1}(s)\xi_i(s))$. Выражение (6) доказано.

Следствие 2. ПЛС-времени выполнения запроса в соответствии с исходным планом (см. рис. 2, а) равно

$$T(s) = T_1(s). \quad (8)$$

Это утверждение следует из определения ПЛС $T_1(s)$ (см. формулу (6)).

Альтернативный план (в виде SQL-запросов — см. рис. 3).

Теорема 3. Производящая функция $W_i(z)$ числа записей, обработанных при выполнении i -го SQL-запроса, определяется следующей формулой:

$$W_i(z) = Q_{i+1}(z)G_i(z)Q_i(z), \quad i = \overline{n, 1}, \quad (9)$$

```
SELECT      An4 as m, fi(An2) as qn INTO TEMPn
```

```
FROM        Rn
```

```
GROUP BY    An4
```

Далее необходимо закодировать $n-2$ SQL-операторов, здесь i изменяется от $n-1$ до 2:

```
SELECT      Ai4 as ri, fi(Ai2) as qi INTO TEMPi
```

```
FROM        Ri, TEMPi+1
```

```
WHERE       Ri.Ai3=TEMPi+1.ri+1 AND Ri.Ai1 = i TEMPi+1.qi+1
```

```
GROUP BY    Ai4
```

Затем должен быть выполнен n -й запрос, который имеет номер 1:

```
SELECT      A10
```

```
FROM        R1, TEMP2
```

```
WHERE       R1.A13=TEMP2.r2 AND R1.A11  $\tau$ 1 TEMP2.q2
```

Рис. 3. Запросы SQL, моделирующие альтернативный план

где

$$Q_{n+1}(z) \equiv 1, \quad Q_1(z) \equiv 1, \quad Q_i(z) = z^{I_{i4}}, \quad G_i(z) = z^{V_i}. \quad (10)$$

Доказательство. Анализ SQL-запросов (см. рис. 3) показывает, что при выполнении каждого i -го запроса, кроме запросов с номерами n и 1, выполняются следующие действия.

1. Читаются записи из таблицы $TEMP_{i+1}$ (ПФ числа записей равна $Q_{i+1}(z)$).

2. Читаются записи из исходной таблицы R_i (ПФ числа записей равна $G_i(z)$). Читаются все записи этой таблицы, так как при выполнении операции группирования $GROUP BY$, определенной в $(i + 1)$ -м запросе, в таблице $TEMP_{i+1}$ сохраняются все значения атрибута связи $A_{i+1,4}$ и в i -м запросе используется условие соединения $A_{i3} = TEMP_{i+1}.ri+1$, где $|ri+1| = |A_{i+1,4}|$.

3. Результат выполнения запроса записывается в таблицу $TEMP_i$ (ПФ числа записей равна $Q_i(z)$). Здесь выполняется операция группирования, число записей в таблице $TEMP_i$ равно мощности атрибута A_{i4} , т.е. величине I_{i4} .

В n -м SQL-запросе (первый запрос, см. рис. 3) поиск и группирование выполняются в одной таблице (таблица $TEMP_{n+1}$ отсутствует), поэтому $Q_{n+1}(z) \equiv 1$. В первом SQL-запросе (последний запрос, см. рис. 3) запись данных во временную таблицу не выполняется, поэтому здесь $Q_1(z) \equiv 1$. Величины V_i и I_{i4} — это параметры, задаваемые проектировщиком (случайные величины с одним состоянием). Это доказывает выражения (10) для $Q_i(z)$ и $G_i(z)$.

Из определения ПФ $Q_{i+1}(z)$, $G_i(z)$, $Q_i(z)$ (см. действия 1–3, рассмотренные выше) и свойства производящих функций получаем равенство (9). Теорема доказана.

Следствие 3. ПФ $W(z)$ числа записей, обработанных в процессе выполнения всех n SQL -запросов (см. рис. 3), равна

$$W(z) = \prod_{i=1}^n W_i(z). \quad (11)$$

Выражение (11) следует из (9) и свойств производящих функций.

В общем случае записи таблицы R_i не упорядочены по атрибуту соединения $Ai3$, поэтому предположим в дальнейшем, что таблицы R_i и $TEMP_{i+1}$ соединяются с использованием метода вложенных циклов NLJ.

Теорема 4. ПЛС $Y_i(s)$ времени выполнения i -го SQL -запроса определяется следующим выражением:

$$Y_i(s) = W_i(\delta_i^*(s)) \cdot G_i(Q_{i+1}(\xi_i^*(s))), \quad i = \overline{n-1}, \quad (12)$$

где ПФ $W_i(z)$ определяется выражением (9); ПФ $G_i(z)$, $Q_i(z)$ определяются выражениями (10);

$$Q_{n+1}(\xi_n^*(s)) \equiv \xi_n^*(s); \quad (13)$$

$\delta_i^*(s)$ — ПЛС-время чтения/записи одной записи базы данных в таблице при выполнении i -го запроса; $\xi_i^*(s)$ — ПЛС-время сравнения записей в соответствии с условием $Ai3 = TEMP_{i+1}.qi+1 \text{ AND } Ai1 \tau_i TEMP_{i+1}.qi+1$ для $i = \overline{n-1}, 1$ и/или определения текущего значения функции агрегирования $f_i(Ai2)$ при успешном сравнении для $i = \overline{n}, 2$.

Доказательство. Первый сомножитель в выражении (12) равен ПЛС-времени ввода/вывода записей в таблицы базы данных при выполнении i -го SQL -запроса. Второй сомножитель равен ПЛС-времени соединения таблиц R_i и $TEMP_{i+1}$ методом NLJ [2], (см. формулу (2.71)) и определения значения функции агрегирования при выполнении i -го запроса. Тогда из свойства ПЛС следует выражение (12). Теорема доказана.

Следствие 4. ПЛС-время выполнения всех n SQL -запросов, т.е. времени выполнения исходного запроса (см. рис. 1) в соответствии с альтернативным планом (см. рис. 3), равно

$$Y(s) = \prod_{i=1}^n Y_i(s). \quad (14)$$

Выражение (14) следует из свойства ПЛС.

Если таблицы R_i и $TEMP_{i+1}$ отсортированы по атрибуту соединения, то СУБД может выбрать метод SMJ для соединения этих таблиц.

В этом случае второй сомножитель в выражении (12) следует заменить на $G_i(\xi_i^*(s))$.

Сравнение времени выполнения запроса для исходного и альтернативного планов. Ранее были получены характеристики времени выполнения запроса с вложенными коррелированными подзапросами и операциями агрегирования (см. рис. 1) для исходного и альтернативного планов (см. рис. 2, *a* и *б*). Далее приведено сравнение математических ожиданий времени выполнения запроса для этих двух планов.

Продифференцировав как сложную функцию правую часть выражения (6) по s в нуле, получим

$$\bar{T}_i = -T'_i(0) = \bar{G} \cdot p_i \cdot (\bar{\delta}_i + \bar{T}_{i+1} + \bar{\xi}_i), \quad i = \overline{1, n}, \quad (15)$$

где $\bar{G}_i = V_i$ — число записей в таблице R_i ; p_i — вероятность, определяемая формулой (3); $\bar{\delta}_i$ — математическое ожидание времени поиска/чтения одной записи i -го уровня вложенности из БД; $\bar{\xi}_i$ — математическое ожидание времени сравнения и определения текущего значения функции агрегирования для i -го уровня, $\bar{T}_{n+1} = 0$.

Из выражения (8) следует, что математическое ожидание времени выполнения запроса, реализуемого в соответствии с исходным планом (см. рис. 2, *a*), равно

$$\bar{T} = \bar{T}_1, \quad (16)$$

где \bar{T}_1 определяется рекуррентной формулой (15).

Продифференцируем выражение (12) по s в нуле:

$$\bar{Y}_i = -Y'_i(0) = \bar{W}_i(\bar{\delta}_i^*) + \bar{G}_i \cdot \bar{Q}_{i+1} \cdot \bar{\xi}_i^*, \quad i = \overline{n, 1}, \quad (17)$$

где

$$\bar{W}_i(\bar{\delta}_i^*) = \bar{Q}_{i+1}(\bar{\delta}_i^*) + \bar{G}_i(\bar{\delta}_i^*) + \bar{Q}_i(\bar{\delta}_i^*); \quad (18)$$

$$\bar{Q}_{i+1}(\bar{\delta}_i^*) = \begin{cases} I_{i+1,4} \cdot \bar{\delta}_i^*, & i = \overline{n-1, 1} \\ 0, & i = n \end{cases};$$

$$\bar{G}_i(\bar{\delta}_i^*) = V_i \cdot \bar{\delta}_i^*;$$

$$\bar{Q}_i(\bar{\delta}_i^*) = \begin{cases} 0, & i = 1 \\ I_{i,4} \cdot \bar{\delta}_i^*, & i = \overline{n, 2} \end{cases};$$

$$\bar{G}_i = V_i;$$

$$\bar{Q}_{i+1} = \begin{cases} I_{i+1,4}, & i = \overline{n-1, 1} \\ 1, & i = n \end{cases}; \quad (19)$$

$\bar{\delta}_i^*$ — математическое ожидание времени чтения/записи одной записи в таблице базы данных при выполнении i -го запроса; $\bar{\xi}_i^*$ — математическое ожидание времени сравнения и определения текущего значения функции агрегирования для i -го запроса.

Выражение (18) следует из соотношений (9) и (10), а выражение (19) — из (10) и (13).

Используя (14), получим выражение для математического ожидания времени выполнения запроса, реализуемого в соответствии с планом на рис. 3:

$$\begin{aligned} \bar{Y} &= -Y'(0) = -\sum_{i=1}^n Y'_i(0) = \sum_{i=1}^n \bar{W}_i(\bar{\delta}_i^*) + \sum_{i=1}^n \bar{G}_i \cdot \bar{Q}_{i+1} \cdot \bar{\xi}_i^* = \\ &= V_n \cdot \bar{\delta}_n^* + \sum_{i=1}^{n-1} (I_{i+1,4}(\bar{\delta}_i^* + \bar{\delta}_{i+1}^*) + V_i \bar{\delta}_i^*) + V_n \cdot \bar{\xi}_n^* + \sum_{i=1}^{n-1} V_i I_{i+1,4} \bar{\xi}_i^* = \\ &= V_n(\bar{\delta}_n^* + \bar{\xi}_n^*) + \sum_{i=1}^{n-1} I_{i+1,4}(\bar{\delta}_i^* + \bar{\delta}_{i+1}^*) + \sum_{i=1}^{n-1} V_i(\bar{\delta}_i^* + I_{i+1,4} \bar{\xi}_i^*). \quad (20) \end{aligned}$$

При выводе формулы (20) учитывали выражения (17), (18) и (19). Если таблицы R_i и $TEMP_{i+1}$ соединяются методом SMJ (см. рис. 3), то в последнем слагаемом выражения (20) величину $I_{i+1,4} \bar{\xi}_i^*$ следует заменить на $\bar{\xi}_i^*$. Формулы (16) и (20) можно использовать для сравнения математических ожиданий времени выполнения исходного запроса, реализованного в соответствии с разными планами (см. рис. 2).

Далее определен подкласс запросов, задаваемый следующими выражениями:

$$\begin{aligned} V_i &= V, \quad \bar{\delta}_i = \bar{\delta}_i^* = \bar{\delta}, \quad \bar{\xi}_i = \bar{\xi}_i^* = \bar{\xi}, \quad i = \overline{1, n}; \\ I_{i+1,4} &= \frac{I_{i4}}{k}, \quad i = \overline{2, n-1}, \quad 1 \leq I_{i4} \leq V, \quad I_{24} = I, \end{aligned} \quad (21)$$

где k — коэффициент изменения мощности атрибута связи при переходе с одного уровня вложенности подзапросов на другой уровень, расположенный ниже.

Полагая, что $\bar{\delta} \gg \bar{\xi}$ и $V\bar{\xi} \gg 2\bar{\delta}$, из формул (16) и (20) можно получить выражения для математических ожиданий времени реализации исходного и альтернативного планов для данного подкласса запросов:

$$\bar{T} = \bar{\delta} \sum_{i=1}^n \frac{V^i}{I^{i-1}} k^{0,5(i-1)(i-2)}, \quad \bar{Y} = \bar{\delta} n V + \bar{\xi} V I \frac{1 - \frac{1}{k^{n-1}}}{1 - \frac{1}{k}}. \quad (22)$$

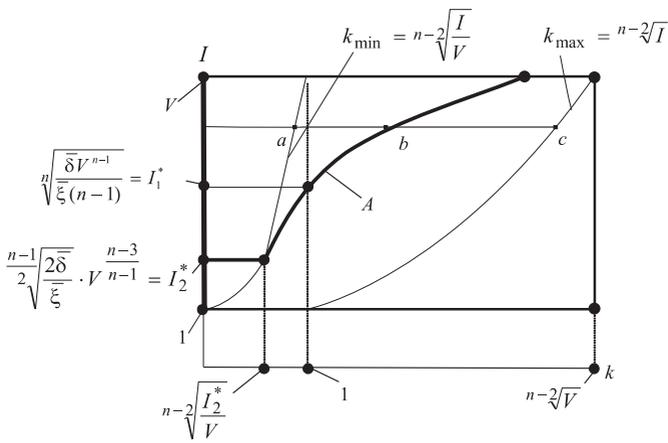


Рис. 4. Области преимущественного использования исходного и альтернативного планов

На рис. 4 построены области преимущественного использования этих планов в пространстве (k, I) (для вариантов n и k).

В случае $n = 2$ времена \bar{T} и \bar{Y} не зависят от k . Поэтому область предпочтительного использования альтернативного плана (см. рис. 2, б; в форме SQL-запросов – см. рис. 3) соответствует неравенству $1 \leq I < I_1^*$, при $I_1^* < I \leq V$ более целесообразно применять исходный план (см. рис. 2, а).

Рассмотрим теперь случай $n > 2$. Кривая A на рис. 4 может быть получена из уравнения $\bar{T} = \bar{Y}$. Кривая A разделяет область $(0 < k \leq n-2\sqrt{V}) \times [1 \leq I \leq V]$ на две части. Нижняя часть соответствует более предпочтительному использованию альтернативного плана (см. рис. 2, б и 3), верхняя часть – более предпочтительному использованию исходного плана (см. рис. 2, а). Для какого-либо I (см. рис. 4) k может изменяться на отрезке $[k_{\min}, k_{\max}]$. На отрезке от точки a до точки b целесообразно использовать исходный план, а на отрезке от b до c – альтернативный план.

Построенные области преимущественного использования планов справедливы, если для соединения таблиц R_i и $TEMP_{i+1}$ (см. рис. 3) применяется метод вложенных циклов NLJ. По аналогии можно построить области и для случая использования метода сортировки слияния SMJ.

Выводы. 1. На основе законов реляционной алгебры разработан метод преобразования исходного плана выполнения запроса с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования в альтернативный план, что позволяет выбирать план с наименьшим временем реализации в СУБД.

2. Доказаны формулы для производящей функции числа обработанных записей и преобразования Лапласа–Стилтгеса времени выполнения запроса для исходного и альтернативного планов, позволяющие

вычислять математические ожидания и моменты более высоких порядков.

3. Для исходного и альтернативного планов выполнено сравнение математических ожиданий времени выполнения запроса из определенного подкласса для различных случаев и построены области преимущественного использования этих планов.

СПИСОК ЛИТЕРАТУРЫ

1. G r a f e G. Query evaluation techniques for large databases // ACM Computing Surveys. – N.Y., 1993. – Vol. 25, № 2. – P. 73–170.
2. Г р и г о р ь е в Ю. А., П л у т е н к о А. Д. Теоретические основы анализа процессов доступа к распределенным базам данных. – Новосибирск: Наука, 2002. – 222 с.
3. Г а р с и а - М о л и н а Г., У л ь м а н Д ж., У и д о м Д ж. Системы баз данных. Полный курс. – М.: Издательский дом “Вильямс”, 2003. – 1088 с.
4. Ш а ш а Д., Б о н н е Ф. Оптимизация баз данных: принципы, практика, решение проблем. – М.: КУДИЦ-ОБРАЗ, 2004. – 432 с.
5. П р и о р и т е т н ы е системы обслуживания / Б.В.Гнеденко, Э.А.Даниелян, Б.Н.Дмитров и др. – М.: Изд-во МГУ, 1973. – 448 с.

Статья поступила в редакцию 24.01.2005



Сергей Петрович Остриков родился в 1962 г., окончил в 1984 г. Минское высшее инженерное зенитное ракетное училище ПВО. Начальник факультета военного обучения МГТУ им. Н.Э.Баумана. Автор ряда работ в области распределенных информационных систем и сетей.

S.P. Ostrikov (b. 1962) graduated from the Minsk Higher School for Antiaircraft and Rocket Engineering. The chief of military training faculty of the Bauman Moscow State Technical University. The author of a number of publications in the field of the distributed information systems and networks.



Юрий Александрович Григорьев родился в 1951 г., окончил в 1975 г. МВТУ им. Н.Э.Баумана. Д-р техн. наук, профессор кафедры систем обработки информации и управления МГТУ им. Н.Э.Баумана. Автор более 60 работ в области распределенных систем обработки данных.

Jury Aleksandrovich Grigoriev (b.1951), graduated from the Bauman Moscow State Technical University in 1975. D.Sc (Eng), professor of faculty of systems of processing of the information and management of Bauman Moscow State Technical University. The author more than 60 works in the field of the distributed systems of data processing.