

НЕ БЫВАЕТ

На правах рукописи

Остриков Сергей Петрович

**Метод анализа процессов доступа к базам
данных с учётом вложенных коррелированных
подзапросов и операций агрегирования**

Специальность 05.13.17 – Теоретические основы информатики

**Автореферат диссертации на соискание ученой степени
кандидата технических наук**



Москва - 2006

[illegible]

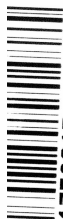
Актуальность темы. В настоящее время в различных информационных системах (ИС) используются системы управления базами данных, поддерживающие реляционную модель данных. Здесь для описания и манипулирования данными применяется стандартный язык SQL, который относится к классу непроцедурных языков. SQL-запрос, поступивший на сервер базы, подвергается оптимизации с целью уменьшения времени его выполнения. Суть оптимизации состоит в генерации нескольких альтернативных планов реализации запроса и выборе плана с наименьшей стоимостью, которая пропорциональна времени выполнения запроса. В теории реляционных баз данных наибольшее внимание уделяется построению оптимальных планов для класса запросов, включающих операции соединения таблиц или имеющих конструкции, которые можно свести к соединению. В теории по существу не рассматривается важный класс запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования. В публикациях, в основном, анализируются запросы с одним коррелированным подзапросом и необоснованно утверждается, что для них оптимальным является план с группированием значений атрибута связи. При этом расчёт стоимости и сравнение альтернативных планов не выполняется. Это привело к тому, что во многих СУБД (Oracle, DB2 и др.) для реализации запросов из указанного класса используются планы на основе операции группирования. Однако, анализ показывает, что эти планы являются оптимальными только в некоторой области изменения параметров запросов и наполнения базы данных. Таким образом, часто для реализации запроса в СУБД выбирается неоптимальный план. Поэтому задача генерации альтернативных планов выполнения запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования, оценки стоимости планов и построения областей их предпочтительного использования является актуальной.

В настоящее время для моделирования информационных систем применяют методы, основанные на знаниях. Средства, поддерживающие эти методы, обеспечивают расчёт характеристик производительности проектируемых систем и их составляющих элементов по описанию запросов и транзакций приложений, таблиц базы данных и её наполнения, топологии сети и виртуальных каналов, параметров сетей и станций. Но в данных системах моделирования можно описывать только те запросы, выполнение которых сводится к соединению подзапросов. Следовательно, доработка ука-

средств, позволяющая учесть особенности обработки запросов с ювневыми вложенными коррелированными подзапросами и операциями агрегирования, а также проведение модельных экспериментов с разработкой рекомендаций при проектировании информационных систем является также актуальной задачей.

Метод анализа

П. С. Стриков



1758015

М Г Т У
И.Н.Э. БИ
БИБЛИОТЕКА

Цель работы. Целью работы является разработка метода анализа и оценки процессов доступа к базам данных для различных планов выполнения запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования, а также моделирование информационных систем с учётом особенностей обработки запросов к базам данных.

В работе решаются следующие **задачи**:

1. Разработка метода преобразования исходного плана выполнения реляционных операций в альтернативный план для запросов из рассматриваемого класса.

2. Разработка метода оценки временных характеристик выполнения указанных планов и построение областей их предпочтительного использования для подкласса запросов.

3. Доработка Комплекса инструментальных Средств Анализа Моделей доступа к базам данных распределённых систем обработки данных (КСАМ), позволяющая учесть особенности обработки запросов из анализируемого класса.

4. Моделирование с помощью пакета КСАМ конкретных информационных систем с учётом запросов к базам данных и их подзапросов, а также выработка рекомендаций для устранения потенциальных «узких мест» в этих проектируемых системах.

Объект исследования. Объектом исследования являются оптимизаторы запросов, обеспечивающие выбор плана с наименьшей стоимостью для реализации запроса к реляционной базе данных.

Предмет исследования. Предметом исследования является процесс выбора рационального плана выполнения запроса с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования

Научная новизна. В работе получены следующие новые научные результаты:

1. На основе законов реляционной алгебры разработан метод преобразования исходного плана выполнения запроса с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования в альтернативный план с операциями группирования значений атрибутов связи.

2. Доказана рекуррентная формула, позволяющая получить производящую функцию числа записей базы данных, обработанных при выполнении запроса в соответствии с исходным планом. Доказана формула для преобразования Лапласа-Стилтьеса времени выполнения запроса в случае применения исходного плана.

3. Доказана формула для производящей функции числа записей, обработанных при выполнении SQL-запроса, обеспечивающего группирование значений атрибута связи. На её основе получена производящая функция числа записей базы данных, обработанных при выполнении запроса в соответствии с альтернативным планом, а также преобразование Лапласа-Стилтьеса времени выполнения запроса в случае применения этого плана.

4. Получены математические ожидания времени выполнения запроса для исходного и альтернативного плана. Для определённого подкласса запросов выполнено сравнение этих математических ожиданий для различных случаев и построены области преимущественного использования этих планов.

Методы исследования. Исследования проводились на основе комплексного использования теории множеств, теории вероятности, теории реляционных баз данных, теории графов, теории экспертных систем.

Практическая ценность полученных результатов. Полученные научные результаты были использованы для доработки экспертной системы КСАМ, обеспечивающей расчёт характеристик производительности проектируемых информационных систем по описанию таблиц базы данных и её наполнения, запросов и транзакций приложений, топологии сети и виртуальных каналов, параметров сетей и станций. Были переработаны фреймы экспертной системы, позволяющие учесть особенности обработки запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования. КСАМ обеспечивает проведение модельных экспериментов с учётом описания запросов к базам данных и их подзапросов с целью выработки рекомендаций при проектировании информационных систем.

Внедрение результатов исследований. Доработанная экспертная система КСАМ была использована при моделировании модулей «Контингент» и «Факультет военного обучения» автоматизированной системы МГТУ им. Н.Э. Баумана на этапе их проектирования. Для каждого модуля выполнен анализ его нагрузочной способности и влияния наполнения базы данных. По результатам моделирования выработаны рекомендации по устранению потенциальных «узких мест» проектируемых систем. В частности, накопление агрегированных данных с помощью триггеров СУБД и хранение этих данных в соответствующих таблицах базы данных позволяет снизить загрузку сервера модуля «Контингент» почти в 200 раз. А изменения схемы базы данных и перепрограммирование трёх критичных запросов приводят к уменьшению загрузки сервера системы «Факультет военного обучения» в 300 раз. Рекомендации были использованы при проектировании указанных модулей АС МГТУ.

Публикации по теме. По материалам работы опубликовано 5 печатных работ.

Апробация работы. Материалы работы были изложены автором на следующих конференциях и семинарах:

1. НТС кафедры ИУ-5 МГТУ им. Н.Э. Баумана, М., 2006.
2. Международный научный симпозиум, «Образование через науку» МГТУ им. Н.Э. Баумана, М., 2005.

Объем работы. Диссертационная работа содержит 158 страниц, 52 рисунка и 12 таблиц, 2 страницы копий актов о внедрении, список литературы из 114 наименований.

Во введении обосновывается актуальность проблемы. Формулируются цели и задачи исследований, приводится перечень основных результатов, выносимых на защиту, и излагается краткое содержание глав диссертации.

В первой главе выполнен обзор моделей доступа к данным, проанализированы различные подходы к оценке показателей качества обработки запросов к базе данных, а также различные приёмы построения альтернативных оптимальных логических планов выполнения запросов.

Рассмотрены модели доступа к данным: файловый сервер, сервер базы данных, сервер приложений, доступ по технологиям Intranet/Internet и в системах с объектной архитектурой. Приведены их преимущества и недостатки. Во всех этих моделях в качестве языка доступа к данным используется SQL (в том числе в объектно-ориентированных СУБД), который относится к классу непроцедурных языков.

Выполнен анализ работ Делиса, Гардарина, Грэфа, Херриса, Мишры, Шапиро, Григорьева Ю.А., Плутенко А.Д. и др., в которых изложены различные подходы к оценке показателей качества обработки запросов к базе данных, используемые при выборе оптимальных планов выполнения запросов в среде СУБД и моделировании систем на основе баз данных:

1. Модели массового обслуживания. Работу СУБД трудно описать в терминах очередей и обслуживающих аппаратов. В предлагаемых моделях не учитываются особенности выполнения приложений, для них сложно получить исходные данные и нелегко проверить выполнение предпосылок использования модели. Тем более в этих моделях не учитывается механизм декомпозиции запросов на подзапросы и особенности их обработки в узлах и сетях информационной системы.

2. Калибруемые аналитические модели. Такие модели отличаются дорогостоящей. Получаемые по результатам натурных экспериментов аналитические выражения справедливы только для данной конфигурации аппаратно-программного комплекса. 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 способ их оценки.

3. Стоимостные модели. В этих моделях рассматриваются, в основном, операции соединения таблиц, приводятся подробные формулы расчёта стоимости для различных физических реализаций соединений. Однако, для таких операций, как агрегирование атрибутов при выполнении вложенных коррелированных подзапросов, такие формулы отсутствуют.

4. Модели, основанные на знаниях. Отличительной особенностью этих моделей является возможность учитывать при моделировании информационных систем описание SQL-запросов. Запросы можно представить на уровне подзапросов и их соединения. Однако, в этих моделях не учитываются особенности выполнения запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования.

Проанализированы различные приёмы построения альтернативных оптимальных логических планов выполнения запросов к базе данных. Показано, что они хорошо проработаны для запросов, где выполняется соединение таблиц или применяются конструкции, которые можно представить в виде операций соединения. Отмечается, что стратегия построения оптимального плана существенно усложняется, если исходный запрос включает многоуровневые коррелированные подзапросы и функции агрегирования атрибутов.

Во второй главе описан рассматриваемый класс запросов к базе данных, разработан метод преобразования исходного плана выполнения запроса в альтернативный план, доказаны выражения, позволяющие оценивать временные характеристики для исходного и альтернативного планов, построены области преимущественного использования этих планов.

На рис. 1 приведено описание исследуемого класса запросов.

```

SELECT A10      FROM R1
WHERE  A11 τ1
      (SELECT f2(A22)      FROM R2
      WHERE  R1.A13=R2.A24 AND A21 τ2
      (SELECT f3(A32)      FROM R3
      WHERE  R2.A23=R3.A34 AND A31 τ3
      (SELECT f4(A42)      FROM R4
      ...
      AND An-1,1 τn-1
      (SELECT fn(An2) FROM Rn
      WHERE  Rn-1.An-1,3=Rn.An4
      ) ...)))

```

Рис. 1. Описание исследуемого класса запросов.

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 $i+1$ -ой таблицей (R_{i+1}) в $i+1$ -ом подзапросе (поэтому под-

запрос является коррелированным). $Ai4$ ($i = \overline{2, n}$) - атрибут i -й таблицы, по которому устанавливается соединение с i -1-ой таблицей (R_{i-1}) в i -ом подзапросе.

На рис. 2а показан результат частичной трансляции описания запроса (см. рис.1) в дерево выражений реляционной алгебры, здесь π и σ - операции проекции и селекции. На каждом i -ом уровне вложенности ($i = \overline{1, n-1}$) введена операция σ без индекса (т.е. без условия), которая определяет отношение, получаемое после выполнения операции селекции с условием $Ai1 \pi_{i+1}(Ai+1,2)$ над отношением $R1$ или $\sigma_{Ai-1,3=Ai4}$. В процессе физической реализации дерево просматривается сверху вниз. Сначала читаются записи из $R1$. Для каждой такой записи определяется отношение $\sigma_{A13=A24}$ (значение $A13$ уже определено). Далее для каждой записи из этого отношения определяется отношение $\sigma_{A23=A34}$ (значение $A23$ уже определено) и так далее до самого нижнего уровня вложенности. Затем для каждой записи из рассматриваемых отношений $\sigma_{Ai-1,3=Ai4}$ выполняется перемещение снизу вверх, связанное с проверкой условий σ (без индекса) и вычислением функций агрегирования π . В работе показано, что исходный логический план может быть реализован несколькими способами: в ядре СУБД и с помощью хранимой процедуры, включающей вложенные курсоры.

Для построения альтернативного логического плана необходимо выполнить следующие шаги на каждом уровне вложенности (см. рис. 2а):

1. Селекцию $\sigma_{Ai-1,3=Ai4}$ переместить вверх по дереву за операцию селекции σ без индекса, т.е. после её выполнения.
2. Операцию селекции $\sigma_{Ai3=Ai+1,4}$ переместить за операцию селекции σ без индекса. На место перемещённой селекции вставить операцию группирования $\gamma(Ai+1,4 \pi_{i+1}, \pi_{i+1}(Ai+1,2) \pi_{i+1})$. Здесь $Ai+1,4$ - это атрибут, по которому выполняется группирование. Группируемое отношение поступает на вход операции γ после выполнения операций, расположенных по дереву ниже. $\pi_{i+1}(Ai+1,2)$ - это операция агрегирования атрибута $Ai+1,2$, которая выполняется для каждой группы. Идентификаторы π_{i+1} и π_{i+1} - это псевдонимы значений, указанных слева от них, т.е. $Ai+1,4$ и $\pi_{i+1}(Ai+1,2)$ соответственно.

3. Операцию селекции σ без индекса и условие $\langle \text{condition} \rangle$ заменить на селекцию $\sigma_{Ai1 \pi_{i+1}}$ над декартовым произведением R_i и γ .

4. Каскад селекций $\sigma_{Ai3=Ai+1,4}$ и $\sigma_{Ai1 \pi_{i+1}}$ заменить на одиночную селекцию $\sigma_{Ai3=\pi_{i+1} \cap Ai1 \pi_{i+1}}$.

Последовательно применяя процедуру 1-4 для каждого уровня вложенности (i меняется от $n-1$ до 1, селекция $\sigma_{A03=A14}$ считается пустой), план, представленный на рис. 2а (исходный план), преобразуется к дереву, которое изображено на рис. 2б (альтернативный план). В диссертации доказано, что преобразования 1-4 являются корректными, т.е. после выполнения снизу вверх процесса, представленного на рис. 2б, будет получен правильный результат поиска данных.

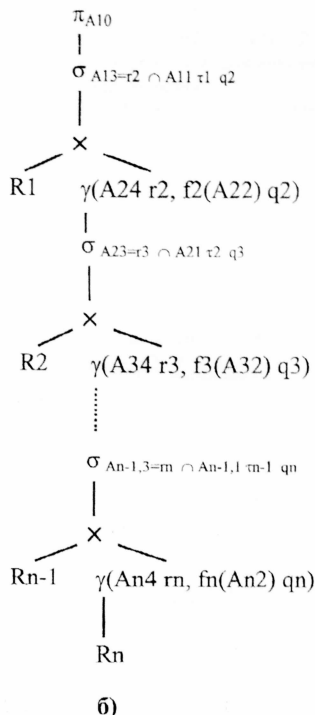
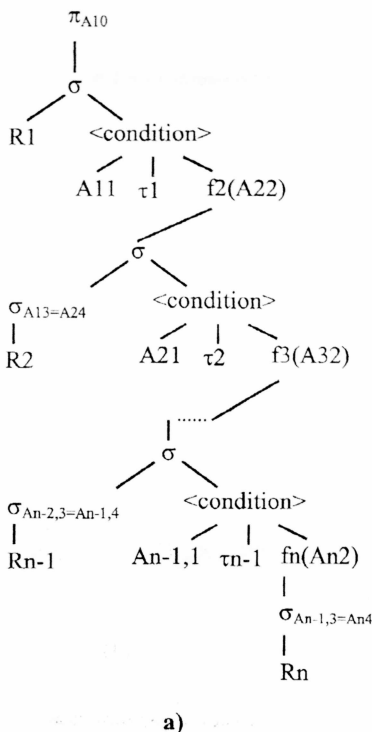


Рис. 2. Исходный план (а) и альтернативный план (б) выполнения запроса.

В некоторых СУБД, например Oracle, план на рис. 2б выполняется неэффективно на физическом уровне (здесь соединение $\sigma_{A_{i3}=r_{i+1} \cap A_{i1} \tau_i q_{i+1}}$ реализуется путём фильтрации всех записей уровня i). Ниже приведены "n" SQL-операторов (рис. 3), совокупность которых соответствует плану на рис. 2б. В каждом запросе выполняется соединение 2-х 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. 2б.

В диссертации доказаны приведённые ниже 4 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.

```

SELECT      An4 as m, fn(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 ti TEMPi+1.qi+1
GROUP BY    Ai4

```

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

```

SELECT      A10
FROM        R1, TEMP2
WHERE       R1.A13=TEMP2.r2 AND R1.A11 t1 TEMP2.q2

```

Рис. 3. Запросы SQL, моделирующие план на рис. 2б.

1. Исходный план.

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

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

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

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 $Ai4$ (число разных значений), $H_{n+1}(z) \equiv 1$.

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

$$H(z) = H_1(z). \quad (4)$$

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

$$T_i(s) = G_i(1 - p_i(1 - \delta_i(s)T_{i+1}(s)\xi_i(s))), \quad i = \overline{1, n}, \quad (5)$$

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

Следствие 2. ПЛС времени выполнения запроса в соответствии с планом на рис. 2а равно

$$T(s) = T_1(s). \quad (6)$$

2. 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:

$$W_i(z) = Q_{i+1}(z)G_i(z)Q_i(z), \quad i = \overline{n,1}, \quad (7)$$

$$\text{где } Q_{n+1}(z) \equiv 1, \quad Q_1(z) \equiv 1, \quad Q_i(z) = z^{I_{i4}}, \quad G_i(z) = z^{V_i}. \quad (8)$$

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

$$W(z) = \prod_{i=1}^n W_i(z). \quad (9)$$

В дальнейшем считается, что таблицы R_i и $TEMP_{i+1}$ (см. рис. 3) соединяются методом NLJ.

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

$$Y_i(s) = W_i(\delta_i^*(s)) \cdot G_i(Q_{i+1}(\xi_i^*(s))), \quad i = \overline{n,1}, \quad (10)$$

где ПФ $W_i(z)$ определяется выражением (7), $Q_{n+1}(\xi_n^*(s)) = \xi_n^*(s)$, ПФ $G_i(z)$, $Q_i(z)$ определяются выражениями (8), $\delta_i^*(s)$ - ПЛС времени чтения/записи одной записи базы данных в таблицу при выполнении i -го запроса, $\xi_i^*(s)$ - ПЛС времени сравнения записей в соответствии с условием $Ai3=TEMP_{i+1}.pi+1 \text{ AND } Ai1 \text{ ti } TEMP_{i+1}.qi+1$ для $i = \overline{n-1,1}$ и/или определения текущего значения функции агрегирования $fi(Ai2)$ при успешном сравнении для $i = \overline{n,2}$.

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

$$Y(s) = \prod_{i=1}^n Y_i(s). \quad (11)$$

Дифференцируя выражения (4) и (9), а также (6) и (11) несколько раз соответственно в точке $z = 1$ и $s = 0$, можно получать моменты высоких порядков числа обработанных записей и времени выполнения запроса. Это важно, так как эти моменты являются исходными данными при моделировании вычислительных систем на макроуровне, т.е. на уровне сети.

Продифференцировав как сложную функцию правую часть выражения (5) по s в нуле, получим

$$\bar{T}_i = -T'_i(0) = \bar{G}_i \cdot p_i \cdot (\bar{\delta}_i + \bar{T}_{i+1} + \bar{\xi}_i), \quad i = \overline{1,n}, \quad (12)$$

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 (6) следует, что математическое ожидание времени выполнения запроса, реализуемого в соответствии с исходным планом (см. рис. 2а), равно

$$\bar{T} = \bar{T}_1, \quad (13)$$

где T_1 определяется рекуррентной формулой (12).

Получено выражение для математического ожидания времени выполнения запроса, реализуемого в соответствии с альтернативным планом в виде SQL-запросов (см. рис. 3):

$$\bar{Y} = -Y'(0) = V_n(\bar{\delta}_n^* + \bar{\xi}_n^*) + \sum_{i=1}^{n-1} I_{i+1,4}(\bar{\delta}_i^* + \bar{\delta}_{i+1}^*) + \sum_{i=1}^{n-1} V_i(\bar{\delta}_i^* + I_{i+1,4}\bar{\xi}_i^*). \quad (14)$$

Используя (13) и (14), можно сравнивать планы по времени реализации и выбирать наилучший из них.

В работе определён подкласс запросов, задаваемый следующими выражениями:

$$V_i = V, \quad \bar{\delta}_i = \bar{\delta}_i^* = \bar{\delta}, \quad \bar{\xi}_i = \bar{\xi}_i^* = \bar{\xi}, \quad i = \overline{1, n}, \quad (15)$$

$$I_{i+1,4} = \frac{I_{i4}}{k}, \quad i = \overline{2, n-1}, \quad 1 \leq I_{i4} \leq V, \quad I_{24} = I,$$

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 $V\bar{\xi} \gg 2\bar{\delta}$, получены выражения для математических ожиданий времени реализации исходного и альтернативного планов для данного подкласса запросов:

$$\bar{T} = \bar{\delta} \sum_{i=1}^n \frac{V^i}{I^{i-1}} k^{\frac{(i-1)(i-2)}{2}}, \quad \bar{Y} = \bar{\delta} nV + \bar{\xi} V I \frac{1 - \frac{1}{k^{n-1}}}{1 - \frac{1}{k}}. \quad (16)$$

В работе рассмотрены различные варианты для n и k и построены области преимущественного использования этих планов в пространстве (k, I) (рис. 4).

В случае $n=2$ время $\bar{T}$ и $\bar{Y}$ не зависят от k (см. (16)). Поэтому область предпочтительного использования альтернативного плана на рис. 2б (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 на рис. 2а.

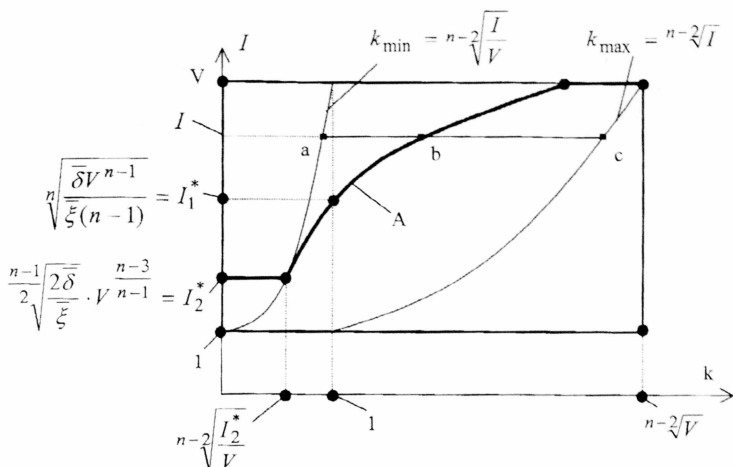


Рис. 4. Области преимущественного использования исходного и альтернативного плана.

Рассмотрим теперь случай $n > 2$. Кривая "А" на рис. 4 может быть получена из уравнения $\bar{T} = \bar{Y}$ (см. (16)). Кривая "А" разделяет область $(0 < k \leq n - 2\sqrt{I/V}) \times [1 \leq I \leq V]$ на две части. Нижняя часть соответствует более предпочтительному использованию альтернативного плана (см. рис. 2б и рис. 3), верхняя часть - более предпочтительному использованию исходного плана (см. рис. 2а). Для какого-либо I (см. рис. 4) k может изменяться на отрезке $[k_{\min}, k_{\max}]$. На отрезке от точки "а" до точки "б" целесообразно использовать исходный план, а на отрезке от "б" до "с" - альтернативный план. Приведённые выше рекомендации по использованию планов справедливы, если таблицы R_i и $TEMP_{i+1}$ (см. рис. 3) соединяются методом NLJ. По аналогии можно построить области предпочтительного использования рассматриваемых планов для метода соединения SMJ.

В третьей главе рассмотрена экспертная система KСAM, которая в отличие от других систем моделирования имеет базу знаний, где хранятся декларативные и процедурные знания о функционировании информационных систем (ИС). Выполнена доработка этой экспертной системы.

Проанализирована функциональная структура KСAM, обеспечивающего моделирование ИС с учётом конфигураций узлов (станций) и сетей, параметров схемы базы данных проектируемой информационной системы, характеристик наполнения БД, спецификаций транзакций на уровне SQL-операторов, параметров распределения таблиц БД и транзакций по узлам ИС. Выполнен анализ схемы базы данных KСAM, предназначенной для хранения исходных данных и результатов моделирования и используемой для доработки этой экспертной системы. Проанализирована база знаний KСAM и машина вывода, осуществляющая навигацию по этой БЗ с целью

анализа описания проектируемой ИС, оценки её характеристик производительности и выявления «узких мест».

Выполнена доработка базы знаний КСАМ, позволяющая моделировать выполнение запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования. Разработан алгоритм, обеспечивающий выбор плана с наименьшей стоимостью в процессе моделирования:

1. Транслировать SQL-запрос в исходное дерево операций и рассчитать число записей, обработанных на i -ом уровне вложенности подзапросов, используя рекуррентную формулу:

$$L_i = L_{i-1} \bar{G}_i p_i, \quad i = \overline{1, n}, \quad (17)$$

где $L_0 = 1$, $p_1 = 1$, $\bar{G}_i = V_i$ - прогнозируемое число записей в таблице R_i , $p_i = 1/I_{i4}$ - вероятность, что запись i -го уровня вложенности удовлетворяет условию связи соседних подзапросов ($Ai3=Ai+1,4$), I_{i4} - мощность атрибута связи, n - количество уровней вложенности подзапросов.

2. Определить число блоков таблицы R_i , которые читаются из базы данных при обработке подзапроса i -го уровня вложенности (для случая, когда индекс кластеризован):

$$N_i = \left\lceil \frac{L_i l_i}{Q_D} \right\rceil, \quad (18)$$

где l_i - длина записи таблицы R_i , Q_D - размер блока диска базы данных, $\lceil \cdot \rceil$ - округление с избытком.

3. Определить число блоков базы данных и числа записей, обрабатываемых в оперативной памяти: $N = \sum_{i=1}^n N_i$, $L = \sum_{i=1}^n L_i$. Оценить стоимость выполнения запроса в соответствии с исходным планом:

$$C_1 = C_{D1} + C_{ПП1} = N \cdot t_B + L \cdot t_3, \quad (19)$$

где C_{D1} , $C_{ПП1}$ - дисковая и процессорная составляющая стоимости, t_B - время на блок, t_3 - время обработки одной записи в ОП.

4. Преобразовать исходное дерево операций в альтернативный план выполнения запроса, используя шаги 1 - 4. Преобразовать план в последовательность SQL-запросов. Рассчитать число записей, которые были прочитаны и записаны в базу данных при выполнении i -го SQL-запроса:

$$L_i = I_{i+1,4} + V_i + I_{i4}, \quad i = \overline{n-1, 1}, \quad L_n = V_n + I_{n4}, \quad \text{где } I_{14} = 0. \quad (20)$$

5. Определить число записей базы данных, обработанных в оперативной памяти при выполнении i -го SQL-запроса:

$$L_i^* = V_i \cdot I_{i+1,4}, \quad i = \overline{n-1, 1}, \quad L_n^* = V_n. \quad (21)$$

6. Определить по формуле (18) число блоков таблицы R_i , которые читаются и записываются в БД при обработке i -го запроса SELECT. Рассчитать число блоков БД и число записей, обрабатываемых в ОП:

$N = \sum_{i=1}^n N_i$, $L^* = \sum_{i=1}^n L_i^*$. Оценить стоимость выполнения запроса в соответствии с альтернативным планом:

$$C_2 = C_{Д2} + C_{ПП2} = N \cdot t_B + L^* \cdot t_3. \quad (22)$$

7. Если $C_1 > C_2$, то для реализации запроса выбрать альтернативный план. Если $C_2 > C_1$, то для реализации запроса выбрать исходный план.

В **четвертой главе** выполнено моделирование модулей «Контингент» и «Факультет военного обучения» автоматизированной системы МЛТУ им. Н.Э. Баумана с учётом запросов к базе данных и выработаны рекомендации по устранению потенциальных «узких мест».

Выполнен анализ структуры, 19 таблиц схемы базы данных и 15 наиболее важных SQL-запросов системы «Контингент», которая функционирует под управлением ОС Linux Red Hat. Многие запросы включают несколько таблиц, которые соответствуют подзапросам, а также операции группирования и агрегирования. В среде доработанной экспертной системы KСAM были описаны указанные параметры модуля «Контингент» и его трехзвенная архитектура «Web-клиент – Web-сервер (Apache) + Сервер приложений (Radical) – Сервер БД (MySQL 4.0)». Здесь также была создана конфигурация узла «Proliant ML530G2T 2P», указаны данные из результатов соответствующего теста TPC-C (отчёт от 8 января 2003 г.). Определён канал сети (Канал 1) между узлами «Сервер приложений» и «Web-клиент (терминал)». Тип сетевого канала – порт коммутатора на 100 Мбит/с, к которому подключён сервер модуля «Контингент». Создана конфигурация канала с пропускной способностью 1000Мбит/с. Определён канал связи (Канал 2) между узлами «Сервер приложений» и «Сервер БД» на 1000Мбит/с. Этот канал моделирует передачу данных между задачами «Сервер приложений» и «Сервер БД» с учётом времени задержки на переключение задач.

С помощью KСAM выполнен анализ нагрузочной способности системы «Контингент» при увеличении числа рабочих станций с 1000 до 3000. Здесь «узким местом» является Сервер БД, его загрузка увеличивается с 0.2 до 0.6 (т. е. загрузка подсистемы ввода/вывода с учётом наличия общесистемного буфера). Были выработаны следующие рекомендации:

1. Загрузка Сервера БД почти равномерно распределена по 5 транзакциям, которые предназначены для получения различных списков. При реализации запроса, соответствующего транзакции, происходит соединение 4 таблиц. Если агрегированные данные (число студентов) не определять каждый раз, а хранить их в записях таблиц Faculty (факультет), Department (кафедра), Group (группа), Student (студент) и использовать триггеры СУБД, то загрузку Сервера БД можно уменьшить почти в 200 раз, что позволяет увеличить нагрузочную способность сервера или расширить его функциональные возможности.

2. Использование коммутатора здесь оправдано, т.к. применение входной сети на концентраторах привело бы к снижению максимального числа рабочих станций с 3000 до 2000.

3. Почти вся загрузка Канала 1 приходится на транзакцию «Студенты в приказе». Анализ соответствующего запроса показывает, что при его выполнении из базы данных читаются и передаются на рабочую станцию значения атрибутов 5 записей таблиц, включённых в запрос. Но в одной из таблиц (Order_student) имеется поле attributes длиной 200 Кбайтов, передача которого и вызывает большую загрузку Канала 1. Уменьшение длины этого поля до 20 Кбайтов позволяет снизить загрузку Канала 1 в 10 раз.

В среде KСAM выполнен анализ влияния наполнения базы данных системы «Контингент» (число записей таблиц было увеличено в 10 раз, что соответствует сохранению информации о студентах, которые завершили обучение в Университете). Загрузка Сервера БД равномерно распределена по 5 транзакциям, которые выполняют чтение различных списков. При реализации каждого из этих запросов выполняется соединение 4 таблиц. Хотя число блоков, обработанных в процессе соединения, возросло с 400 до 4000 (т.е. в 10 раз), загрузка сервера увеличилась с 0.2 до 0.4 (т.е. только в 2 раза). Это объясняется тем, что число дисков, на которых хранится база данных, возросло в 5 раз (процент заполнения диска зафиксирован на уровне 30%). Распределение данных по разным дискам особенно важно для таблицы Student, так как основная доля соединяемых записей читается из этой таблицы.

Выполнен анализ структуры, 17 таблиц схемы базы данных и 25 SQL-запросов ИС «Факультет военного обучения» (ФВО), функционирующей под управлением Windows 2000 Server. Наиболее критичные запросы включают несколько таблиц, которые соответствуют подзапросам, а также операции группирования и агрегирования. Эти параметры ИС ФВО, её трёхуровневая архитектура «Web-клиент - Web-сервер (Internet Information Services, ASP-страницы) - Сервер БД (MS SQL Server 2000 Standard Edition)», а также конфигурация узла «Proliant ML530-X1000-1P» были описаны в среде экспертной системы KСAM. По аналогии с модулем «Контингент» определены канал сети Канал 1 и канал связи Канал 2.

С помощью KСAM выполнен анализ нагрузочной способности ИС ФВО при увеличении числа рабочих станций с 200 до 600. Здесь «узким местом» является Сервер БД, его загрузка увеличивается с 0.3 до 0.9. Были предложены следующие рекомендации:

1. Удалить таблицу Group_Student из запросов и включить в таблицу Student индексированный атрибут GroupID.

2. Хранить число студентов в каждой записи таблицы Group (поле StudentNumber), для поддержки агрегированных данных использовать триггеры СУБД.

3. Изменить следующие запросы:

ВК.Список групп - поиск осуществлять в одной таблице Group,

ВК.Список студентов - соединять 4 таблицы, при этом на соединяемые записи таблицы Student накладывается условие $s.GroupID = @GroupID$.

НУК.Список студентов - поиск осуществлять в одной таблице Student, в запросе наложить условие $s.GroupID = @GroupID$.

Реализация указанных рекомендаций позволяет сократить число читаемых блоков почти в 300 раз и во столько же раз уменьшить загрузку Сервера БД.

В среде КСАМ выполнен анализ влияния наполнения базы данных ИС ФВО (число записей в таблицах Group_Student, Student и Student_Mark было увеличено в 3 раза). При росте числа записей количество блоков, обработанных при выполнении критичных запросов ВК.Список групп, ВК.Список Студентов, НУК.Список Студентов увеличилось с 1600 до 4800 на каждый запрос, т.е. в 3 раза. Во столько же раз возросла загрузка Сервера БД: с 0.3 до 0.9. Это свидетельствует о том, при изменении наполнения таблиц количество дисков, где хранится база данных, осталось прежним (процент заполнения одного диска был принят равным 50). Для рассматриваемого случая было рекомендовано снизить долю заполнения диска базы данных до 25%.

Существенное увеличение производительности модулей АС МГТУ в результате корректировки запросов и схемы БД объясняется тем, что проектировщик, кодируя SQL-запросы, как правило, не знает, как они выполняются в среде СУБД (SQL является непроцедурным языком). Поэтому моделирование с помощью доработанного КСАМ наиболее критичных SQL-запросов помогает ему определить наиболее эффективный способ доступа к базе данных.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИОННОЙ РАБОТЫ

1. Разработан новый метод оценки и выбора планов реализации запросов с вложенными коррелированными подзапросами и операциями агрегирования, и на его основе определены процедурные знания, позволяющие учитывать этот класс запросов при моделировании информационных систем.

2. Разработан метод преобразования исходного плана выполнения произвольного запроса из рассматриваемого класса в альтернативный план, что позволяет выбирать план с наименьшим временем реализации. Доказана эквивалентность этих логических планов с точки зрения конечных результатов поиска данных. Показано, что рассмотренные планы имеют несколько физических реализаций.

3. Доказана рекуррентная формула, позволяющая получить производящую функцию числа записей базы данных, обработанных при выполнении запроса в соответствии с исходным планом. Доказана формула для преобразования Лапласа-Стилтьеса времени выполнения запроса в случае применения исходного плана.

4. Доказана формула для производящей функции числа записей, обработанных при выполнении SQL-запроса, обеспечивающего группирование значений атрибута связи. На её основе получена производящая функция числа записей базы данных, обработанных при выполнении запроса в соответствии с альтернативным планом, а также преобразование Лапласа-Стилтьеса времени выполнения запроса в случае применения этого плана.

5. Для исходного и альтернативного плана выполнено сравнение математических ожиданий времени выполнения запроса из определённого подкласса для различных случаев и построены области преимущественного использования этих планов. Разработан алгоритм, обеспечивающий выбор плана с наименьшей стоимостью.

6. Полученные результаты были использованы для доработки экспертной системы КСАМ, обеспечивающей расчёт характеристик производительности проектируемых информационных систем с учётом особенностей обработки запросов с многоуровневыми вложенными коррелированными подзапросами и операциями агрегирования.

7. Доработанная экспертная система КСАМ была использована при моделировании модулей «Контингент» и «Факультет военного обучения» автоматизированной системы МГТУ им. Н.Э. Баумана на этапе их проектирования с целью выявления и устранения потенциальных «узких мест». Проведённый анализ показал, что накопление агрегированных данных с помощью триггеров СУБД и хранение этих данных в соответствующих таблицах базы данных позволяет снизить загрузку сервера модуля «Контингент» почти в 200 раз. А изменения схемы базы данных и перепрограммирование трёх критичных запросов приводит к уменьшению загрузки сервера системы «Факультет военного обучения» в 300 раз.

Основное содержание диссертации отражено в следующих работах:

1. Григорьев Ю.А., Остриков С.П. Выбор оптимального плана выполнения запроса с вложенными коррелированными подзапросами // Информатика и системы управления (Благовещенск). – 2006. – № 1. – С.3-14.

2. Григорьев Ю.А., Остриков С.П., Плутенко А.Д. Моделирование информационной системы «Контингент» // Информатика и системы управления (Благовещенск). – 2006. – № 1. – С.15-26.

3. Григорьев Ю.А., Остриков С.П. Оценка времени выполнения запросов с коррелированными подзапросами и операциями агрегирования // Вестник МГТУ. Приборостроение. - 2006. №1 – С. 98-110.

4. Клепиков В.И. Остриков С.П., Использование CALS-технологий в производстве и эксплуатации бортовых систем управления // Вестник МГТУ. Приборостроение. – 2002. – №1 – С. 85-92.

5. Жарова И.З., Остриков С.П., Меняев Я.А. Особенности администрирования ЛВС в образовательных учреждениях // Проблемы построения и эксплуатации систем обработки информации и управления: Сборник статей. / МГТУ им. Н.Э. Баумана. – 2000.– Вып. 1. – С.78–86.