



На правах рукописи

Гребенников Николай Андреевич

**Метод поиска оптимального плана
выполнения запросов к базам данных
на основе нисходящей стратегии**

**Специальность 05.13.17 – Теоретические основы информатики
(технические науки)**

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

A handwritten signature in black ink, consisting of stylized, overlapping loops and strokes.

Москва - 2004

Работа выполнена в Московском государственном техническом университете им. Н. Э. Баумана.

Научный руководитель: доктор технических наук,
профессор Григорьев Ю.А.

Официальные оппоненты: доктор технических наук,
профессор Федулов А.А.

кандидат технических наук
Жуков И.В.

Ведущая У1661290 проблем информатики
Гребенников Н.А. (академии наук (г. Москва)
Метод поиска оптимального
Задачу плана выполнения запросов к
заседания базам данных на основе » февраль 2004 года на
государственный нисходящей стратегии 212.141.10 в Московском
0-00 с имени Н.Э. Баумана по

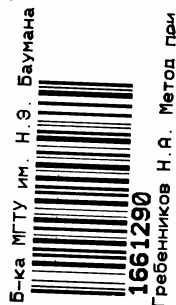
У 1661290

библиотеке

ВОЗВРАТИТЕ КНИГУ НЕ ПОЗЖЕ
обозначенного здесь срока

ателью, просим

ОВ
Гираж



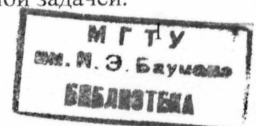
502

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В настоящее время при разработке информационных систем широко используются системы управления базами данных (СУБД) на основе реляционной модели данных. Составной частью этой модели являются методы оптимизации запросов, позволяющие анализировать множества альтернативных планов выполнения запросов и выбирать оптимальные планы их реализации.

Разработка эффективного оптимизатора запросов в составе СУБД остаётся одной из наиболее важных задач. Это позволяет существенно уменьшить время выполнения запросов в системах разных классов. Например, при принятии решений в системах организационного управления выполняются SQL-запросы, соединяющие несколько десятков таблиц. При этом существующие оптимизаторы в основном используют эвристические методы, что может приводить к выбору неоптимального плана выполнения запросов. Часто запросы выполняются часами. Другой класс систем, системы оперативной обработки информации, характеризуются большим числом клиентов и наличием дорогостоящей техники, обрабатывающей большой поток запросов. Уменьшение времени выполнения запросов для таких систем позволяет увеличить нагрузочную способность и сократить число единиц дорогостоящей техники.

Практика показывает, что подходы, используемые для оптимизации простых запросов в системах оперативной обработки информации, оказываются не столь эффективными для сложных аналитических запросов, выполняемых в системах организационного управления. Причиной этого является то, что восходящая стратегия оптимизации, разработанная более 15 лет назад для оптимизатора системы System R и не претерпевшая с тех пор значительных изменений, имеет один принципиальный недостаток, который ограничивает её применение для оптимизации сложных запросов: перед генерацией первого плана некоторой группы, оптимизируются все её дочерние группы. При этом нет возможности избежать генерации мультывыражений в группах, которые не являются частью оптимального плана, на основе информации, получаемой с верхнего уровня. В последние годы была предложена нисходящая стратегия оптимизации, позволяющая преодолеть недостаток восходящего подхода за счёт использования верхних и нижних границ стоимости для отсеечения групп планов. Это делает её более перспективной для дальнейшего развития, однако в существующей реализации нисходящей стратегии оптимизации вероятность срабатывания механизма отсеечения групп невелика, что обесценивает её преимущества. Поэтому разработка метода поиска оптимального плана, позволяющего использовать преимущества нисходящей стратегии и уменьшить время анализа пространства поиска за счёт сокращения числа рассматриваемых альтернативных планов, является актуальной задачей.



Цель работы. Целью работы является разработка новых методов поиска оптимального плана на основе нисходящей стратегии анализа альтернативных планов и использование полученных теоретических результатов при разработке оптимизатора запросов к СУБД.

В работе решены следующие **задачи**:

- 1) выполнен критический анализ существующих методов оптимизации выполнения запросов,
- 2) разработаны методы поиска оптимального плана с использованием нисходящей стратегии анализа альтернативных планов,
- 3) разработан оптимизатор запросов, реализующий предложенные методы нисходящей стратегии поиска,
- 4) проведён анализ эффективности предлагаемых методов поиска оптимального плана с помощью разработанного оптимизатора запросов,
- 5) выполнено исследование процесса обработки запросов в реальной вычислительной системе на этапе её реорганизации.

Объект исследования. Объектом исследования настоящей работы являются оптимизаторы запросов, встраиваемые в современные системы управления базами данных с целью увеличения производительности выполнения запросов.

Предмет исследований. Предметом исследований настоящей работы являются процессы оптимизации запросов пользователей системами управления базами данных.

Научная новизна. В работе получены следующие новые научные результаты:

- 1) разработан новый метод поиска оптимального плана с использованием нисходящей стратегии анализа альтернативных планов, позволяющий уменьшить время поиска за счёт рационального выбора верхней и нижней границ стоимости выражений в мето-структуре,
- 2) предложен метод генерации пространства поиска, который позволяет определить все логически эквивалентные мультिवыражения, описывающие альтернативные планы выполнения запроса, и исключить все планы, содержащие декартовы произведения,
- 3) получены формулы для оценки вероятности отсекаания планов, содержащих декартовы произведения,
- 4) выведены формулы для оценки числа альтернативных деревьев поиска и необходимых для их представления мультिवыражений в мето-структуре для различных типов деревьев поиска и топологий запросов,
- 5) получены формулы для оценки стоимости операции соединения таблиц базы данных методом вложенных циклов и методом сортировки и слияния, учитывающие наличие и типы индексов по столбцам соединения, а также объём доступной оперативной памяти;

Методы исследования. Исследования проводились на основе комплексного использования теории реляционных баз данных, теории множеств, теории вероятности, теории графов, комбинаторики и теории формальных языков.

Практическая ценность полученных результатов. В работе для практического использования полученных результатов разработан оптимизатор запросов, реализующий предложенные методы и алгоритмы нисходящей стратегии поиска. Оптимизатор позволяет задавать различные наборы логических и физических операторов, правил генерации и ограничений пространства поиска, анализировать эффективность использования индексов по различным атрибутам таблиц. Предложена методика использования оптимизатора для прогнозирования времени выполнения запросов к СУБД. Он может быть также использован для оценки времени выполнения запросов в реальных вычислительных системах на этапах реорганизации и модернизации.

Внедрение результатов исследований. Разработанный оптимизатор запросов использован в качестве программного компонента при проектировании специальной системы управления базами данных для миникомпьютера Сайбико, что позволило уменьшить время выполнения запросов в СУБД в среднем на 40%.

Разработанные методы, методика и оптимизатор запросов использованы в процессе исследования процесса обработки запросов в системе «Cybiko PDBS» на этапе её реорганизации с целью прогнозирования характеристик функционирования системы при добавлении к ней подсистемы поддержки принятия решений. Проанализированы планируемые запросы к СУБД, учитывающие сложный характер задач указанной подсистемы и содержащие большое количество операторов соединения, агрегации и сортировки. В результате получены оценки времени выполнения этих запросов к СУБД. Для двух запросов получено время выполнения, превышающее установленный десятисекундный предел. Для устранения этих «узких мест» было предложено использовать индексы по столбцам соединения и сортировки. Анализ оптимизированного варианта системы показал, что время выполнения всех тестовых запросов не превышает установленных в требованиях ограничений.

Публикации по теме работы. По материалам работы опубликовано 7 печатных работ.

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

- 1) Теоретическая конференция компании Сайбико, М., 2002.
- 2) НТС кафедры ИУ-5, МГТУ им. Н.Э. Баумана, М., 2003.

Объем работы. Диссертационная работа содержит 183 страницы, 65 рисунков и 18 таблиц, 1 страницу копий актов о внедрении, список литературы из 162 наименований. Основной текст изложен на 157 страницах.

СОДЕРЖАНИЕ РАБОТЫ

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 «Анализ существующих методов построения оптимизаторов запросов» приведено описание процесса обработки запросов в СУБД, выделены и проанализированы компоненты ядра СУБД, участвующие в обработке запросов пользователей. Приведена общая классификация подходов к оптимизации запросов, реализуемых в современных СУБД. Проведён анализ основных подходов к оптимизации запросов с точки зрения реализуемых стратегий поиска оптимального плана выполнения запросов. Выполнен анализ существующих методов оценки стоимости операции соединения таблиц базы данных.

Выделены шесть этапов обработки запроса: лексический и синтаксический анализ текста запроса, логическая оптимизация, получение набора альтернативных планов выполнения запроса, выбор оптимального плана из набора альтернативных планов, формирование представления оптимального плана и выполнение этого представления. Выделены компоненты СУБД, участвующие в обработке запросов: модуль разбора запроса, модуль выбора стратегии обработки запроса и модуль выполнения запроса.

Проведён анализ процесса оптимизации запросов, выделены три основные понятия, определяющие процесс оптимизации запроса: пространство поиска, стоимостная модель и стратегия поиска. В соответствии с выделенными понятиями сформулированы три признака и проведена классификация существующих подходов к процессу оптимизации запросов, реализуемых в современных СУБД.

Выделены и проанализированы существующие стратегии поиска, реализуемые оптимизаторами запросов. Показано, что существующие методы имеют ряд недостатков и ограничений:

1. Восходящий алгоритм оптимизатора System R имеет ограниченную расширяемость и не пригоден для запросов, включающих соединения более 15 отношений.
2. Восходящий расширяемый алгоритм оптимизатора Starburst использует эвристики на фазе преобразований, что может приводить к выбору неоптимального плана,
3. Нисходящий подход, используемый в генераторах оптимизаторов Exodus и Volcano, имеет невысокую эффективность, так как все логические выражения генерируются перед генерацией физических выражений. Поэтому Volcano генерирует $O(3^N)$ выражений.
4. Нисходящий расширяемый подход Cascades является наиболее перспективным из представленных выше. Однако он также имеет невы-

сокую производительность, так как в существующей реализации вероятность срабатывания механизма отсеечения групп невелика.

Применяя ту или иную стратегию поиска, оптимизатор запросов формирует множество планов выполнения запросов, для сравнения которых он использует некоторую стоимостную функцию. В работе проведён анализ существующих методов оценки стоимости операции соединения методами вложенных циклов, сортировки и слияния, хеш-соединения. В диссертации проанализированы работы Грэфа, Хэрриса, Мишры, Шапиро и выявлены достоинства и недостатки рассмотренных методов. Показано, что для операции хеш-соединения таблиц известны формулы расчёта стоимости для различных случаев. Однако для двух других физических реализаций операции соединения (методов вложенных циклов, сортировки и слияния) ни в одном методе оценки стоимости не учитывается наличие и тип доступных индексов, которые в значительной степени влияют на время выполнения операции соединения.

Проведённый анализ показал необходимость разработки нового метода поиска оптимального плана на основе нисходящей стратегии анализа альтернативных планов и использования более совершенной стоимостной модели.

Во второй главе «Методы повышения эффективности нисходящей стратегии поиска» проведен анализ алгоритма нисходящей стратегии поиска, выполнено сравнение алгоритмов нисходящей и восходящей стратегий поиска. Разработаны новые методы поиска оптимального плана на основе нисходящей стратегии, позволяющие увеличить вероятность отсеечения групп. Доказаны лемма и теорема, определяющие эффективный метод генерации пространства поиска с использованием нисходящей стратегии поиска. Получены формулы для оценки числа альтернативных деревьев поиска и необходимых для их представления мультивыражений в мето-структуре для различных типов деревьев поиска и топологий запросов. На их основе проведена оценка предлагаемых методов поиска на основе нисходящей стратегии. Предложена стоимостная модель для операции соединения таблиц базы данных методом вложенных циклов и методом сортировки и слияния.

Алгоритм нисходящей стратегии поиска оптимального плана основывается на применении специального внутреннего представления пространства поиска. Для описания этого представления в работе рассмотрены концепции логического и физического операторов, операторного выражения, мультивыражения, группы и мето-структуры. *Логический оператор* – это функция, отображающая входы оператора на его выходы. *Физический оператор* – это алгоритм, позволяющий получить выходные кортежи на основе кортежей его входов. *Операторное выражение (выражение)* – это дерево операторов, в котором операторы с нулевым числом входов выступают в качестве листьев узлов дерева (они определяют доступ к хранимым отношениям), а операторы с ненулевым числом входов – в качестве внутренних узлов. При этом входы каждого оператора являются

выходами его дочерних операторов. Выражение называется логическим или физическим в зависимости от типа оператора верхнего уровня. Начиная с оптимизатора Volcano, пространство поиска в нисходящих оптимизаторах обозначается термином *мето-структура*. Мето-структура состоит из двух взаимно рекурсивных структур данных, которые называются мультивыражениями и группами. *Мультивыражение* – это оператор, входами которого являются группы (примером мультивыражения является соединение двух групп отношений). По определению все иерархически связанные выражения с одинаковым оператором верхнего уровня и одинаковыми входами этого оператора представляются одним мультивыражением. *Группа* – это класс эквивалентных мультивыражений, позволяющих получить один и тот же результат (одинаковые выходные наборы кортежей).

В работе показано, что основным преимуществом нисходящей стратегии поиска над восходящей является возможность использования механизма отсечения групп при оптимизации запроса. *Отсечение группы* – ситуация, при которой группа не оптимизируется, т.е. для неё не генерируется ни одного мультивыражения в процессе оптимизации. Отсечение групп приводит к уменьшению числа мультивыражений, рассматриваемых в процессе оптимизации (сокращению пространства поиска) и, следовательно, к уменьшению времени оптимизации.

В работе проведён стоимостной анализ алгоритма нисходящей стратегии поиска (рис. 1). Анализ показал, что условием, определяющим возможность отсечения группы, является то, что текущая стоимость рассматриваемого мультивыражения группы ($C_{нг(i)}$) превышает текущую верхнюю границу стоимости группы ($C_{вг}$). При этом стоимость $C_{нг(i)}$ в момент сравнения включает стоимость базового оператора мультивыражения.

```

(1)  $C_{опт}(k, \text{группа}, C_{вг})$  (сначала для корневой группы  $C_{вг} = \infty$ )
(2) Цикл по физическим мультивыражениям группы (по  $i$ )
{
(3)  $C_{нг(i)}$  = стоимость базового оператора  $i$ -ого физического мультивыражения.
(4) Если  $C_{нг(i)} > C_{вг}$ , то переход к  $i+1$ 
(5) Цикл по входным группам  $j$  физического мультивыражения  $i$ 
{
(6) Получение стоимости оптимального плана  $j$ -ого входа:
 $C_{опт} = C_{опт}(i, \text{вход } j, C_{вг} - C_{нг(i)})$ .
(7) Если  $C_{опт} = -1$  (оптимального плана не существует), то переход к  $i+1$ .
(8)  $C_{нг(i)} = C_{нг(i)} + C_{опт}$ .
}
(9) Если  $C_{нг(i)} < C_{вг}$ , то  $C_{вг} = C_{нг(i)}$ .
}

```

Рис. 1. Стоимостной анализ алгоритма нисходящей стратегии поиска

Указанное условие будет выполняться тем чаще, чем больше $C_{нг(i)}$ и чем меньше $C_{вг}$. Поэтому в работе разрабатываются два новых метода поиска оптимального плана на основе нисходящей стратегии.

В работе предложен метод увеличения значения стоимости мультивыражения ($M1$) в момент сравнения за счёт учёта информации о входах,

которые были оптимизированы ранее, и использования логических свойств для других входов. На рис. 2 показаны изменения в алгоритме нисходящей стратегии поиска (см. рис. 1), реализующие метод M1.

Предложенные изменения позволяют увеличить значение $C_{нг}(i)$, путём добавления в строках (3б) и (3с) стоимостей тех входных групп, которые могут быть получены без проведения их оптимизации. Если сумма этих стоимостей будет превышать верхнюю границу стоимости группы, то нет необходимости оптимизировать оставшиеся входные группы.

(3а)	$C_{нг}(i)$ = стоимость базового оператора МЕхрг +
(3б)	стоимость входов, уже имеющих оптимальные планы для требуемых свойств +
(3с)	стоимость копирования результатов остальных входов
...	
(5а)	Для каждого оптимизируемого входа МЕхрг, для которого ранее не был найден оптимальный план для требуемых свойств {
...	
(8а)	$C_{нг}(i)$ = $C_{нг}(i)$ + стоимость оптимального плана входа – стоимость копирования результатов входа }

Рис. 2. Метод увеличения нижней границы стоимости группы (M1)

Для тех входных групп, для которых оптимальные планы были получены ранее (при поиске в другом поддереве), их вычисленные стоимости могут использоваться без повторной оптимизации этих групп. Предварительное значение стоимости для какой-либо оставшейся неоптимизированной входной группы представляет собой стоимость передачи результирующего набора кортежей к следующему оператору в дереве запроса. Эта стоимость может быть оценена на основе выбранной стоимостной модели и размеров результатов групп без их оптимизации, так как размер результирующего отношения является логическим свойством группы и одинаков для всех мультिवыражений в группе.

Если вычисленная таким образом стоимость мультिवыражения превышает текущую верхнюю границу стоимости группы, то оставшиеся входные группы текущего мультिवыражения не оптимизируются. В противном случае необходима оптимизация оставшихся входных групп. Для этой цели организуется цикл по всем входным группам текущего мультिवыражения, чьи оптимальные планы не были найдены ранее (строка (5а) на рис. 2). В цикле выполняется поиск оптимального плана входной группы, и далее из значения стоимости текущего мультिवыражения исключается стоимость копирования результата входной группы, которая была добавлена ранее (строка (3с)), и замещается полной стоимостью оптимального плана этой входной группы, которая включает в себя и стоимость копирования результата (строка (8а)).

В работе предложен также метод генерации пространства поиска (M2), при использовании которого первые мультिवыражения групп в этом пространстве будут иметь меньшую стоимость базового оператора. Это обеспечивает уменьшение верхней границы стоимости $C_{вг}$ на ранней стадии оптимизации группы. Основой метода является использование исход-

ного лево-глубокого операторного дерева $R = [A_1, \dots, A_{k-1}] \bowtie [A_k]$ и применение двух правил генерации пространства поиска: коммутативности и правой ассоциативности соединений:

УП-1: 1) мультिवыражение $u \bowtie x$ добавляется в группу $[xu]$ с $x \bowtie u$,

2) для $u \bowtie x$ запрещается применение правил УП-1 и УП-2.

УП-2: 1) мультिवыражение $x \bowtie (yz)$ добавляется в группу $[xyz]$ с $(xy) \bowtie z$,

2) для $x \bowtie (yz)$ запрещается применение правила УП-2,

3) создаётся новая группа $[yz]$ с первым мультिवыражением $u \bowtie z$,

4) для $u \bowtie z$ разрешается применение правил УП-1 и УП-2,

5) при этом x, y – группы, z – однотабличная группа.

Шаги метода генерации пространства поиска для запроса соединения k таблиц:

1. Создать группу верхнего уровня $[A_1, \dots, A_k]$, включив в неё одно мультिवыражение $[A_1, \dots, A_{k-1}] \bowtie [A_k]$.
2. Применить правила УП-1 и УП-2 к данному мультिवыражению.
3. Применить правила УП-1 и УП-2 к первым мультिवыражениям новых групп, созданных на шаге 2 с помощью правила 3) УП-2.

В работе доказана следующая лемма, показывающая, что когда входное операторное дерево является лево-глубоким, набор правил УП-1, УП-2 позволяет генерировать все возможные эквивалентные логические мультिवыражения в пространстве поиска.

Лемма 1 (полнота набора правил). Пусть R – исходное лево-глубокое операторное дерево. Предположим, что для R применяется предлагаемый в настоящей работе метод генерации пространства поиска (правила УП-1 и УП-2). Тогда справедливы следующие утверждения: (1) каждая оптимизированная группа будет содержать все возможные эквивалентные логические мультिवыражения, (2) если группа содержит более одной таблицы, то второй вход первого мультिवыражения группы будет однотабличной группой, (3) новые группы будут создаваться только путём применения правила ассоциативности.

Лемма 1 показывает, что порядок оптимизации можно изменять так, что первое мультिवыражение в каждой группе будет иметь однотабличную группу в качестве правого входа. Однако левый вход в общем случае может включать декартово произведение отношений.

В работе доказана следующая теорема, показывающая, что использование метода М2 для оптимизации связанных запросов позволяет исключить из рассмотрения планы, включающие декартовы произведения.

Теорема 1 (качество метода). Пусть Q – это связанный ациклический запрос и его исходное операторное дерево R является лево-глубоким. Пусть для R применяется предлагаемый выше метод генерации пространства поиска (правила УП-1 и УП-2). Тогда каждая связанная группа в результирующем пространстве поиска будет начинаться с мультिवыражения, содержащего хотя бы один план без декартовых произведений.

В работе проведена оценка предлагаемых методов поиска оптимального плана на основе нисходящей стратегии. При этом использовалась следующая предпосылка: применение предлагаемых методов поиска приводит к отсечению всех несвязанных групп в пространстве поиска. Для оценки числа мультिवыражений в несвязанных группах пространства поиска был проведён анализ размеров пространства поиска. В рамках этого анализа получены формулы для оценки числа альтернативных деревьев поиска и необходимых для их представления мультिवыражений в метоструктуре для различных типов деревьев поиска и топологий запросов. Результаты анализа показаны в табл. 1.

Число мультिवыражений в мето-структуре **Табл. 1.**

Топология запроса/Тип дерева	Кустовое дерево	Лево-глубокое дерево
Полностью соединённый	$3^n - 2^{n+1} + n + 1$	$n2^{n-1}$
Линейный	$(n^3 + 2n)/3$	n^2
Звездообразный	—	$(n-1)2^{n-2} + 2n - 1$

На основе полученных формул рассчитаны вероятности отсечения групп для случаев кустового и лево-глубокого деревьев поиска и линейной и звездообразной топологий запросов:

1. *Кустовое дерево поиска / линейный запрос:*

$$P_{\text{отс}} = (N_{\text{пс}} - N_{\text{лин}}) / N_{\text{пс}} = (3^n - 2^{n+1} - \frac{n^3 - 6n + 1}{3}) / (3^n - 2^{n+1} + n + 1). \quad (1)$$

2. *Лево-глубокое дерево поиска / линейный запрос:*

$$P_{\text{отс}} = (N_{\text{пс}} - N_{\text{лин}}) / N_{\text{пс}} = (2^{n-1} - n + 1) / 2^{n-1}. \quad (2)$$

3. *Лево-глубокое дерево поиска / звёздообразный запрос:*

$$P_{\text{отс}} = (N_{\text{пс}} - N_{\text{лин}}) / N_{\text{пс}} = ((n+1)2^{n-2} - 2n + 1) / (n2^{n-1}). \quad (3)$$

В работе проведён расчёт вероятностей для запросов, включающих от 3 до 13 отношений. Полученные значения вероятностей (0,5 для звездообразных запросов и почти 0,9 для линейных) позволяют говорить о высокой эффективности разработанных методов поиска оптимального плана.

Целью любой стратегии поиска является выбор оптимального плана выполнения запроса среди множества альтернативных планов. При этом для оценки этих планов используются некоторые стоимостные модели. Стоимость плана зависит от стоимостей составляющих его физических операторов. Известно, что операция соединения таблиц в процессе обработки запросов является наиболее ресурсоёмкой. В работе предложена стоимостная модель, в рамках которой получены формулы для оценки стоимости операции соединения двух таблиц базы данных методом вложенных циклов и методом сортировки и слияния, учитывающие наличие и типы индексов по столбцам соединения, а также объём доступной оперативной памяти.

В работе получены формулы оценки стоимости процессорных ин-

струкций и дискового ввода-вывода для метода вложенных циклов в случаях: (1) файлового сканирования обеих таблиц, (2) файлового сканирования внешней таблицы и доступа по кластеризованному или некластеризованному хеш-индексу или В⁺-индексу к внутренней таблице. Для метода сортировки и слияния стоимостные оценки получены в случаях: (1) отсутствия индексов, наличия (2) кластеризованного или (3) некластеризованного В⁺-индекса по столбцу соединения внешней таблицы. Для каждого из трёх последних случаев рассмотрены варианты отсутствия индексов и наличия кластеризованного или некластеризованного В⁺-индекса по столбцу соединения внутренней таблицы.

В качестве примера рассмотрим формулы для оценки стоимости операции соединения двух таблиц методом вложенных циклов для случая файлового сканирования внешней таблицы R и доступа по кластеризованному хеш-индексу внутренней таблицы S. В этом случае стоимость процессорных инструкций определяется следующей формулой:

$$C_{CPU} = T(R) * (C_{hash} + \frac{T(S)}{B(A, S)} * C_{comp}), \quad (4)$$

где $T(R)$, $T(S)$ – число кортежей в отношениях R и S соответственно, $B(A, S)$ – число разделов в хеш-таблице по атрибуту A отношения S, C_{hash} , C_{comp} – число инструкций процессора для хеширования ключа и сравнения ключей в оперативной памяти соответственно.

Стоимость ввода-вывода зависит от числа доступных буферов памяти b . Если b не превышает среднее число страниц в каждом хеш-разделе, т.е. $b < (1 + \lfloor \frac{P(S)}{B(A, S)} \rfloor)$, то для S выделяется только один буфер и формула стоимости ввода-вывода имеет вид:

$$C_{I/O} = P(R) + T(R) * (1 + \lfloor \frac{P(S)}{B(A, S)} \rfloor), \quad (5)$$

где $P(R)$, $P(S)$ – число страниц в отношениях R и S соответственно.

В случае же, когда $b \geq (1 + \lfloor \frac{P(S)}{B(A, S)} \rfloor)$, для S выделяется m буферов размера $1 + \lfloor \frac{P(S)}{B(A, S)} \rfloor$, где m – наибольшее целое такое, что $b \geq m * (1 + \lfloor \frac{P(S)}{B(A, S)} \rfloor)$.

Другими словами, буфера разделяются на m частей, каждая из которых может содержать все страницы среднего раздела. Тогда доступ к страницам таблицы S может рассматриваться как случайный доступ к разделам по схеме испытаний Бернулли с вероятностью “успеха” (доступа к странице на диске), равной $p = (N-s)/N$. Таким образом, число страниц, считанных с диска в результате « k » обращений к файлу размера N при использовании s буферов памяти, может быть оценено как:

$$C_{I/O} = p * k = \frac{N-s}{N} * k. \quad (6)$$

Используя это выражение и подставляя в него значения $N = B(A, S)$, $k = T(R)$ и $s = m$, получаем стоимость ввода-вывода для случая, когда $b = m * (1 + \left\lfloor \frac{P(S)}{B(A, S)} \right\rfloor)$:

$$C_{i/o} = P(R) + (1 + \left\lfloor \frac{P(S)}{B(A, S)} \right\rfloor) * \left\lfloor \frac{B(A, S) - m}{B(A, S)} \right\rfloor * T(R). \quad (7)$$

Аналогичным образом в работе получены формулы для оценки стоимости операции соединения для других случаев.

В третьей главе «Разработка оптимизатора запросов с нисходящей стратегией поиска» предложена формализованная модель процесса оптимизации запросов, на основе этой модели разработаны архитектура и логический проект оптимизатора запросов, реализующего предложенные методы нисходящей стратегии поиска, предложена методика использования оптимизатора запросов для прогнозирования времени выполнения запросов СУБД.

В работе обоснована необходимость построения обобщённой модели процесса оптимизации, которая позволяет объединить разрозненные данные, необходимые для построения такого сложного программного продукта, как оптимизатор запросов. Разработанная модель процесса оптимизации включает модели логических и физических операций, стоимостную модель, модель правил, модель ограничений пространства поиска, а также применение некоторой стратегии поиска для выбора оптимального плана:

$$\text{МПО} = \{\text{МЛО}, \text{МФО}, \text{СМ}, \text{МП}, \{\text{СП}, \text{МОПП}\}\}$$

Модель логических операций (МЛО) определяет модели базы данных и операторов, используемых для описания всех возможных запросов пользователя. Она включает логические операторы и логические свойства. В модели выделены следующие основные логические операторы: чтение отношения, селекция, эквисоединение, проекция, группировка, сортировка и др.

Модель физических операций (МФО) определяет набор операций, реализуемых модулем выполнения запросов СУБД. Она содержит физические операторы для реализации всех операторов модели логических операций, а также два физических свойства: стоимость выполнения и упорядоченность кортежей. В модели рассматриваются следующие физические операторы: файловое и индексное сканирование, соединение методом вложенных циклов, методом сортировки и слияния, хеш-соединение, быстрая сортировка и др. Для каждого физического оператора в работе приведён алгоритм его выполнения.

Модель правил (МП) определяет набор правил преобразований в пространстве поиска. В работе разработаны три группы правил: 10 правил, определяющих преобразования в логическом пространстве поиска, 12 правил, определяющих правила замены логических операторов на реализующие их физические операторы, и 2 дополнительных правила.

Стоимостная модель (СМ) представляет собой набор аналитических формул для расчёта стоимости физических операторов. Для каждого физического оператора МФО вычисляется стоимость выполнения процессорных инструкций и стоимость операций ввода-вывода. Формулы для оценки стоимостей получены на основе анализа алгоритмов выполнения каждой физической операции. При этом используются два типа параметров – мощности входных и выходных таблиц и параметры стоимостной модели, определяющие время выполнения элементарных операций. С помощью этих параметров выполняется привязка стоимостной модели к параметрам аппаратного обеспечения компьютера, на котором работает СУБД.

Применение правил и вычисление стоимостей в процессе оптимизации контролируется *стратегией поиска* (СП), определяющей, какие планы в пространстве поиска следует рассматривать и оценивать, и в какой последовательности. При этом для уменьшения времени поиска оптимизатор может применять методы *ограничения пространства поиска* (МОПП), описанные в соответствующей модели.

В соответствии с предложенной моделью процесса оптимизации запросов разработана архитектура оптимизатора и определены форматы входных и выходных данных. Основные компоненты оптимизатора показаны на рис. 3.

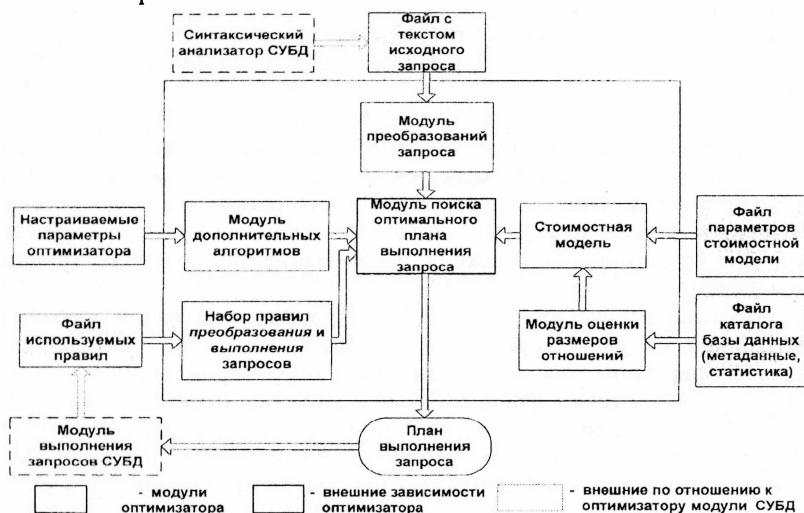


Рис. 3. Структура оптимизатора запросов

Центральным компонентом оптимизатора является модуль поиска оптимального плана, реализующий нисходящую стратегию поиска. В соответствии с подходом, впервые предложенным в оптимизаторе Volcano, алгоритм поиска в разработанном оптимизаторе разделён на

несколько частей, называемых задачами. Оптимизация запроса начинается с запуска задачи оптимизации группы верхнего уровня. В процессе выполнения она порождает задачи оптимизации более мелких групп. Все задачи помещаются в стек и выполняются в основном цикле оптимизации. В работе выделены и спроектированы алгоритмы пяти отдельных задач модуля поиска, позволяющие оптимизировать группы, физические и логические мультिवыражения, применять правила и исследовать группы. Задача оптимизации физического мультिवыражения реализует предложенные методы поиска оптимального плана M1 и M2.

В работе предложена методика использования разработанного оптимизатора для исследования процесса обработки запросов в реальных вычислительных системах на этапах реорганизации и модернизации. Для методики определены область её применения и процедура использования. Она включает следующие шаги:

- 1) определение используемых моделей оптимизации,
- 2) выбор стратегии поиска и ограничений пространства поиска,
- 3) описание параметров прикладной программной среды (системного каталога базы данных и исполняемых запросов),
- 4) применение выбранной стратегии поиска для нахождения оптимального плана выполнения запроса,
- 5) использование полученной стоимости оптимального плана в качестве оценки времени выполнения запроса.

В работе предложены два способа применения разработанной методики, отличающихся реализацией первых двух шагов. Первый способ (более простой) включает процедуру настройки набора правил и параметров стоимостной модели, а также выбор методов ограничения пространства поиска из списка доступных. Этот способ подходит для анализа процесса выполнения запросов в закрытых коммерческих СУБД. В случае наличия данных о внутреннем устройстве используемой СУБД может быть использован второй способ применения разработанной методики. Он включает разработку программного кода, учитывающего особенности конкретной СУБД такие, как логические и физические операторы, правила генерации и выполнения. Такая настройка обеспечивается модульным построением расширяемых оптимизаторов и демонстрирует их преимущества.

Для получения адекватных результатов с помощью разработанной методики необходимо выполнить оценку параметров стоимостной модели для конкретной аппаратной платформы сервера СУБД. В работе предложен способ оценки этих параметров, который основывается на методах линейного регрессионного анализа.

В **четвертой главе** «Использование разработанного оптимизатора запросов для анализа эффективности предложенных методов поиска оптимального плана и при разработке программных продуктов компании Сайбико» приведены результаты исследования эффективности предложенных методов поиска на основе нисходящей стратегии, опи-

сано использование оптимизатора запросов при разработке специальной СУБД для миникомпьютера Сайбико, а также рассмотрены результаты исследования системы «Cybiko PDBS» с помощью разработанного оптимизатора запросов.

Целью первого исследования являлось количественное определение выигрыша, получаемого при использовании разработанных методов, по сравнению с базовым алгоритмом нисходящей стратегии, используемым в коммерческих СУБД. Эксперименты показали высокую эффективность предложенных методов поиска оптимального плана и подтвердили полученные теоретические результаты: выигрыш по числу генерируемых мультивыражений составил около 60% для звездообразных запросов и более 90% – для линейных запросов.

В работе проведён анализ адекватности результатов, получаемых с помощью разработанного оптимизатора, с использованием тестов TPC-H международной ассоциации Transaction Processing Performance Council. Для проведения экспериментов были выбраны 6 запросов, включающие операции соединения большого числа таблиц, а также операции сортировки и агрегации. В экспериментах использовались результаты теста TPC-H, полученные для системы HP900 Superdome Enterprise Server с СУБД Oracle 9i. В соответствии с аппаратными параметрами указанной платформы была произведена настройки оптимизатора запросов, после чего с его помощью было оценено время выполнения запросов теста TPC-H. Сравнение полученных результатов и результатов тестов TPC-H позволяет говорить о том, что погрешность оценки времени выполнения запросов с помощью разработанного оптимизатора составляет не более 30-35%.

Разработанный оптимизатор запросов был использован в качестве программного компонента при проектировании специальной системы управления базами данных для миникомпьютера Сайбико. Для оценки эффективности, достигаемой за счёт использования оптимизатора запросов, проведены эксперименты, в которых выполнено 30 тестовых запросов с использованием и без использования разработанных методов поиска на основе нисходящей стратегии. Эксперименты показали, что использование разработанных методов позволяет уменьшить время выполнения запросов к СУБД в среднем на 40%.

На последнем этапе выполнения работы разработанный оптимизатор запросов был использован для анализа процесса выполнения запросов в системе «Cybiko PDBS». Указанная система установлена и эксплуатируется в московском и тайваньском офисах компании Сайбико и предназначена для сбора, хранения и анализа информации по производству миникомпьютеров. При этом в тайваньском офисе происходит занесение информации в базу данных системы, а в московском офисе – анализ получаемых данных отделом маркетинга. На этапе перепроектирования системы было принято решение о добавлении к ней новой подсистемы поддержки принятия решений. Для выявления «узких мест» и поиска возможных вариантов оптимизации функционирования

этой подсистемы в случае прогнозирования неудовлетворительных результатов был использован разработанный оптимизатор запросов.

В рамках проведённого исследования выполнено описание каталога базы данных системы, выделены и описаны пять основных исполняемых запросов, настроены параметры стоимостной модели на основе анализа аппаратной конфигурации сервера СУБД. Использование оптимизатора, реализующего предложенные методы поиска оптимального плана, позволило спрогнозировать время выполнения выделенных запросов и выявить потенциальные проблемы в анализируемой подсистеме. Было установлено превышение ограничения времени реакции для запросов получения полной информации об устройстве и получения среднемесячной загрузки производственных линий.

Для устранения этих «узких мест» предложено решение, основанное на использовании индексов по столбцам соединения и сортировки. Это позволяет использовать другой алгоритм соединения таблиц и уменьшить время сортировки за счёт более быстрого индексного доступа к данным. Анализ модернизированного варианта системы показал, что время выполнения всех тестовых запросов не превышает установленных в требованиях к системе ограничений.

ЗАКЛЮЧЕНИЕ

1. Предложен новый метод поиска оптимального плана с использованием нисходящей стратегии анализа альтернативных планов, позволяющий уменьшить время поиска за счёт рационального выбора верхней и нижней границ стоимости мультивыражений в анализируемых группах отношений.
2. Разработан метод генерации пространства поиска, который позволяет определить все логически эквивалентные мультивыражения, описывающие альтернативные планы выполнения запроса, и исключить все планы, содержащие декартовы произведения.
3. Выведены формулы для оценки числа альтернативных деревьев поиска и необходимых для их представления мультивыражений для различных типов деревьев поиска и топологий запросов. На их основе получены формулы для оценки вероятности отсеечения планов, содержащих декартовы произведения.
4. Получены формулы для оценки стоимости операций соединения таблиц базы данных методом вложенных циклов и методом сортировки и слияния, учитывающие наличие и тип индексов по столбцам соединения, а также объём доступной оперативной памяти.
5. На основе предложенных методов поиска оптимального плана разработан оптимизатор запросов, позволяющий задавать различные логические и физические операторы, правила генерации и ограничения пространства поиска, а также анализировать эффективность использования индексов по различным атрибутам таблиц и прогнозировать время выполнения запросов СУБД.
6. С помощью разработанного оптимизатора запросов выполнен анализ

эффективности предложенных методов поиска оптимального плана, который показал, что метод отсеечения групп гарантирует получение оптимального результата в отличие от эвристических методов и имеет перед ними преимущества по времени поиска в случае линейной топологии исходного запроса.

7. Разработанный оптимизатор запросов использован в качестве программного компонента при проектировании специальной системы управления базами данных для миникомпьютера Сайбико, что позволило уменьшить время выполнения запросов в СУБД в среднем на 40%.
8. Проведено исследование процесса обработки запросов в системе Sybiko PDBS на этапе её реорганизации с целью прогнозирования характеристик функционирования системы при добавлении к ней подсистемы поддержки принятия решений. Выявлены запросы, время выполнения которых превышает установленные ограничения, и предложены методы устранения этих «узких мест», основанные на использовании индексов по столбцам соединения и сортировки.

Основное содержание диссертации отражено в следующих работах:

1. Григорьев Ю.А., Гребенников Н.А. Методы поиска оптимального плана на основе нисходящей стратегии. // Информатика и системы управления (Благовещенск). – 2003. – №1(5). – С. 3-15.
2. Постников В.М., Гребенников Н.А. Стратегии поиска плана оптимизатора запросов СУБД // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение (М.). – 2003. – №2. – С. 102-124.
3. Гребенников Н.А., Постников В.М. Разработка метода и модели оценки времени выполнения запросов пользователей серверами современных СУБД // Информатика и системы управления (Благовещенск). – 2002. – №2(4). – С. 12-24.
4. Постников В.М., Гребенников Н.А. Формализация процесса обработки запросов пользователей в системе управления базами данных ORACLE // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение (М.). – 2002. – № 3. – С. 55–71.
5. Гребенников Н.А., Постников В.М. Подход к разработке формализованной схемы для исследования работы современных промышленных СУБД // Проблемы построения и эксплуатации систем обработки информации и управления (М.). – 2002. – №4. – С. 53–64.
6. Постников В.М., Гребенников Н.А. Технология обработки запросов пользователей в СУБД ORACLE // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение (М.). – 2001. – № 2. – С. 106–126.
7. Постников В.М., Гребенников Н.А. Анализ процессов обработки SQL-запросов серверами промышленных СУБД // Проблемы построения и эксплуатации систем обработки информации и управления (М.). – 2000. – №2. – С. 152–160.