УДК 004.043
DOI 10.25205/1818-7900-2018-16-2-19-30

И. В. Бельченко, **Р. А. Дьяченко**

Кубанский государственный технологический университет
ул. Московская, 2, Краснодар, 350072, Россия

ilur@mail.ru, emessage@rambler.ru

МЕТОДИКА ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ НЕБОЛЬШИХ ИНФОРМАЦИОННЫХ СИСТЕМ ЗА СЧЕТ ОПТИМАЛЬНОЙ РЕСТРУКТУРИЗАЦИИ ДАННЫХ НА ОСНОВЕ МНОГОМОДАЛЬНОГО РАСПРЕДЕЛЕНИЯ АТРИБУТОВ

Рассматривается системный подход к повышению производительности небольших информационных систем за счет оптимальной реструктуризации табличных структур данных. Авторами сформулирована задача оптимизации количества информационных блоков, необходимых для выполнения группы запросов на считывание информации, предложена целевая функция и структурные ограничения. Проанализирована невозможность использования грубых методов поиска оптимального решения. Предложена методика многомодального распределения атрибутов в зависимости от частоты появления в группе запросов. Проведен эксперимент, подтверждающий эффективность разработанной методики для небольших информационных систем.

Ключевые слова: система поддержки принятия решений, оптимизация, структуры данных, базы данных, системный анализ.

Введение

Большинство многопользовательских информационных веб-систем выделяется такими требованиями, как оперативное взаимодействие с пользователем [1]. Эффективное исполнение данного требования зависит не только от аппаратной составляющей, включающей в себя серверное оборудование и линии связи, но и от реализации программных компонентов, среди которых программное приложение, реализованное с применением веб-технологий и система управления базой данных (СУБД). В статье рассмотрена методика повышения производительности информационной системы, за счет уменьшения среднего времени выполнения группы запросов на чтение информации базы данных. От структуры данных, способах ее физического размещения на жестких дисках зависит количество обращений к дисковым накопителям, которые сопровождаются соответствующими прерываниями и задержками по времени [2].

Важным понятием при рассмотрении вопроса физической организации баз данных является понятие блока. Блок – это минимальный адресуемый элемент внешней памяти, с помощью которого осуществляется обмен информацией между оперативной и внешней памятью. Запись и чтение блоков осуществляется через буферную часть оперативной памяти. Для организации каждого файла базы данных в зависимости от его размера во внешней памяти выделяется от одного до N блоков, где размещаются записи. В одном блоке могут разместиться все записи или в нескольких блоках одна запись, или в одном блоке одна запись. От этого

Бельченко И. В., *Дьяченко Р. А.* Методика повышения производительности небольших информационных систем за счет оптимальной реструктуризации данных на основе многомодального распределения атрибутов // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 19–30.

будет зависеть время считывания и записи элементов файла. Записи в блоках размещаются плотно, без промежутков, последовательно. В блоке часть памяти отводится под служебную информацию: относительный адрес свободных участков памяти, указатели на следующий блок и т. д. Для хранения поступающих данных, которые должны размещаться в одном блоке, заполненном уже полностью, выделяется дополнительный блок памяти в области переполнения записи, организованной в виде одного блока, где записи связываются указателями в одну цепь.

Таким образом, на скорость поиска влияют: объем блока в байтах, объем файла, количество записей в блоке файла, количество записей в блоке индекса, количество блоков в файле, доля резервной части блока, число полей в записи, размер записи в байтах [2].

Постановка задачи вертикальной реструктуризации табличных структур данных

Процесс построения оптимальной модели данных информационной системы включает оптимальное вертикальное распределение таблиц базы данных по блокам на дисковом накопителе. Основным критерием оптимизации модели данных информационной системы является минимальный размер строки таблицы реляционной базы данных, позволяющий в одном блоке хранить больше данных и, как следствие, минимизировать количество операций чтения блоков данных с жесткого диска при выполнении запросов к базе данных. Это достигается за счет уменьшения объема данных, побочно участвующих в запросе [3].

В рамках методики предлагается разделить таблицы базы данных на несколько сущностей, связанных отношением один к одному. В соответствии с принципами блочного хранения данных в СУБД каждая таблица будет храниться в отдельном наборе блоков. При выполнении запроса на чтение информации СУБД считывает блоки данных с жесткого диска в оперативную память каждой таблицы, атрибуты которой участвуют в запросе.

Задача повышения производительности информационной системы сводится к поиску оптимального разделения табличных структур базы данных с учетом конкретной группы запросов на чтение информации, выявленной статистически в рамках жизненного цикла БД [4].

Оптимизация табличных структур данных информационной системы

Для формализации задачи рассмотрим множества и параметры, влияющие на скорость обработки запросов на чтение информации к исследуемой таблице базы данных.

1. Целочисленный параметр TS, равный количеству атрибутов в исследуемой таблице.

2. Вектор типов данных $DBT = \left\{ dbt_{idbt} \middle| idbt = \overline{1, ndbt} \right\}$, которые поддерживаются конкретной выбранной СУБД. Элемент вектора – занимаемый элементом типа размер данных в байтах памяти.

3. Набор атрибутов (столбцов таблицы) TA, который задан бинарной матрицей, элемент которой $ta_{ita,\,jta}$ равен единице, если столбец ita таблицы имеет тип jta, $ita = 1,...,TS$, $jta = 1,...,ndbt$.

4. Множество, представляющее группу запросов $Q = \left\{ q_{iq} \middle| iq = \overline{1, nq} \right\}$ на чтение информации из таблицы базы данных, элемент множества – кортеж из двух элементов $q_{iq} = \left\{ SFQ_{iq}, QA_{iq} \right\}$, где SFQ_{iq} – числовой параметр, равный частоте появления запроса за выбранный период времени, $QA_{iq} = \left\{ qa_{iqa} \middle| iqa = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS. $qa_{iqa} = 1$, если атрибут таблицы TA участвует в за-

просе, и 0 в противном случае. nq – количество запросов в статистической выборке, выявленной в рамках жизненного цикла БД.

5. Множество индексов, характеризующихся набором полей таблицы, по которым построен индекс $IN = \left\{ in_{iin} \middle| iin = \overline{1, nin} \right\}$. Элемент множества $in_{iin} = \left\{ in_{iin,jin} \middle| jin = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS, $in_{iin,jin} = 1$, если атрибут jin таблицы TA участвует в индексе in_{iin}, и 0 в противном случае.

6. Хранимые процедуры и функции $PF = \left\{ pf_{ipf} \middle| ipf = \overline{1, npf} \right\}$, характеризующиеся набором полей, используемых в теле хранимой процедуры или функции. Элемент множества $pf_{ipf} = \left\{ pf_{ipf,jpf} \middle| jpf = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS, $pf_{ipf,jpf} = 1$, если атрибут jpf таблицы TA участвует в теле хранимой процедуры или функции pf_{ipf}, и 0 в противном случае.

7. Множество триггеров базы данных $TG = \left\{ tg_{itg} \middle| itg = \overline{1, ntg} \right\}$, характеризующихся набором полей таблицы, используемых в теле триггера. Элемент множества $tg_{itg} = \left\{ tg_{itg,jtg} \middle| jtg = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS, $tg_{itg,jtg} = 1$, если атрибут jtg таблицы TA участвует в теле триггера tg_{itg}, и 0 в противном случае.

Анализ влияния количества физических блоков данных, используемых таблицей базы данных, на общее время выполнения группы запросов к ней

Множество запросов Q к рассматриваемой таблице обрабатывается СУБД за время $T(Q, TA, DBT)$. Временные затраты $T(Q, TA, DBT)$ можно представить в виде суммы временных затрат на чтение блоков данных таблиц $T_h(Q, TA, DBT)$, участвующих в запросах Q, и остальных временных затрат $T_o(Q, TA, DBT)$, к которым относятся временные затраты на выполнение плана обработки запроса, на передачу информации и т. д.:

$$T(Q, TA, DBT) = T_h(Q, TA, DBT) + T_o(Q, TA, DBT).$$

В рамках методики предлагается уменьшить слагаемое, влияющее на общее время выполнения запроса $T_h(Q, TA, DBT)$. Временные затраты $T_h(Q, TA, DBT)$ в общем виде зависят от количества операций чтения блоков данных таблиц с жесткого диска. Пусть временная задержка, связанная со считыванием одного блока данных равна T_b, тогда

$$T_h(Q) = \left(\sum_{iq}^{nq} B(q_{iq}, TA, DBT) \right) \cdot T_b,$$

где $B(q_{iq}, TA, DBT)$ – число информационных блоков, которые необходимо считать с жесткого диска в кэш СУБД для дальнейшего выполнения запроса q_{iq} к таблице, заданной бинарной матрицей TA. Кэш СУБД находится в оперативной памяти вычислительного устройства.

Функция $B(q_{iq}, TA, DBT)$ вычисляется как отношение:

$$B\left(q_{iq},TA,DBT\right)=\frac{RC\cdot RS\left(q_{iq},TA,DBT\right)}{DB},$$

где

RC – количество строк в рассматриваемой таблице;

DB – фиксированный размер блока данных выбранной СУБД (в большинстве СУБД он равен 8Кб);

$RS\left(q_{iq},TA,DBT\right)$ – величина, характеризующая дисковое пространство, занимаемое одной строкой таблицы в байтах:

$$RS\left(q_{iq},TA,DBT\right)=RSS\left(q_{iq},TA,DBT\right)+RST\left(q_{iq},TA,DBT\right).$$

Здесь

$RSS\left(q_{iq},TA,DBT\right)$ – количество памяти, занимаемое служебными отметками СУБД для строки, считываемое при выполнении запроса q_{iq};

$RST\left(q_{iq},TA,DBT\right)$ – количество памяти, занимаемое атрибутами таблицы в строке, считываемое при выполнении запроса q_{iq}.

Параметры RC и DB остаются неизменными.

Так как временную задержку T_b, связанную со считыванием одного блока данных, допускается считать постоянной величиной, на сумму временных затрат на чтение блоков данных таблицы TA – $T_h\left(Q,TA,DBT\right)$ влияет количество блоков, необходимое для считывания, которое вычисляется как функция

$$F\left(Q,TA,DBT\right)=\sum_{iq}^{nq}B\left(q_{iq},TA,DBT\right).$$

Подставим в формулу $F\left(Q,TA,DBT\right)$, формулу функции $B\left(q_{iq},TA,DBT\right)$. Функция, определяющая количество блоков, необходимых для считывания с жесткого диска в оперативную память при выполнении множества запросов Q к рассматриваемой таблице *TA*:

$$F\left(Q,TA,DBT\right)=\sum_{iq}^{nq}\left(\frac{RC\cdot RS\left(q_{iq},TA,DBT\right)}{DB}\right).$$

Методика уменьшения количества физических блоков данных, используемых СУБД для выполнения группы запросов к таблице, за счет ее оптимального разделения на дочерние таблицы

В рамках методики предлагается разделить рассматриваемую таблицу на $NB\in\left[1;TS\right]$ дочерних таблиц, связанных с родительской отношением один к одному, 1:1.

Введем следующую переменную:

$$x_{ij}=\begin{cases}1, & \text{если } j\text{-й атрибут нужно выделить в } i\text{-ю таблицу,}\\ 0 & \text{в противном случае.}\end{cases}$$

Переменные представляют собой бинарную матрицу для таблицы реляционной базы данных размерностью $TS \times TS$, где TS – количество атрибутов таблицы. Строки матрицы соответствуют таблицам, на которые разбивается родительская таблица, а столбцы соответствуют их атрибутам.

Количество блоков $BM(Q, RC, DB, DBT, TA, X)$, которое необходимо считать с жесткого диска для выполнения множества запросов Q к таблице TA, вычисляется как функция, равная сумме блоков, которые необходимо считать с жесткого диска для выполнения множества запросов Q к каждой из дочерних таблиц. Максимальное количество дочерних таблиц равно числу атрибутов родительской таблицы TA и равно NB.

$$BM(Q, RC, DB, DBT, TA, X) =$$
$$= \sum_{iq=1}^{nq} \left| \frac{RC \cdot RSM(DBT, TA, X_1) \cdot FQ(q_{iq}, X_1)}{DB} \right| +$$
$$+ \sum_{iq=1}^{nq} \left| \frac{RC \cdot RSM(DBT, TA, X_{irs}) \cdot FQ(q_{iq}, X_{irs})}{DB} \right| +$$
$$+ \sum_{iq=1}^{nq} \left| \frac{RC \cdot RSM(DBT, TA, X_{nb}) \cdot FQ(q_{iq}, X_{nb})}{DB} \right|,$$

где

$$RSM(DBT, TA, X_{irs}) = \left(\sum_{j}^{TS} \left[x_{irs,j} \cdot \left(\sum_{idbt}^{ndbt} DBT_{idbt} \cdot TA_j \right) \right] + RDS(DBT, TA, X_{irs}) \right),$$

$$FQ(q_{iq}, X_{irs}) = \begin{cases} q_{iq}(SFQ), & \text{если } \sum_{u}^{TS} (X_{irs})_u \cdot (q_{iq}(QA))_u > 0, \\ 0 & \text{в противном случае.} \end{cases}$$

$$irs = 1, ..., nb, \quad j = 1, ..., TS, \quad idbt = 1, ..., ndbt.$$

Параметры RC и DB являются постоянными, RSM – функция, характеризующая количество информации, которая занимает одну строку дочерней таблицы irs в байтах, $RDS(DBT, TA, X_{irs})$ – функция, характеризующая дисковое пространство, занимаемое служебными отметками СУБД в строке дочерней таблицы irs в байтах. Следовательно, задача повышения производительности системы сводится к поиску такого разделения таблицы на дочерние, при котором сумма блоков, которые необходимо считать в кэш СУБД для выполнения множества запросов Q, минимально.

Целевая функция

$$\min_{x_{irs,j}} \sum_{iq,irs} \left| \frac{\left(\left(\sum_{j}^{TS} \left[x_{irs,j} \cdot \left(\sum_{idbt}^{ndbt} dbt_{idbt} \cdot ta_j \right) \right] + RDS(DBT, TA, x_{irs}) \right) \right) \cdot RC \cdot FQ(q_{iq}, x_{irs})}{DB} \right|,$$

где

$$FQ\left(q_{iq},x_{irs}\right)=\begin{cases}q_{iq}\left(SFQ\right), & \text{если } \sum\limits_{u}^{TS}\left(x_{irs}\right)_u\cdot\left(q_{iq}\left(QA\right)\right)_u>0,\\ 0 & \text{в противном случае.}\end{cases}$$

$$irs=1,...,nb,\quad j=1,...,TS,\quad idbt=1,...,ndbt.$$

При структурных ограничениях:

1) каждый атрибут родительской таблицы может присутствовать только в одной дочерней таблице

$$\sum_{m_1}x_{m_1,i_1}=1,\quad m_1=1,...,TS,\quad i_1=1,...,TS;$$

2) атрибуты таблицы, используемые при построении индексов, должны принадлежать хотя бы одной дочерней таблице

$$\forall ix,\quad ix=1,...,\left|IN\right|:\prod_{m_2}^{TS}\left[\sum_{i_2}^{TS}\left(x_{m_2,i_2}\cdot in_{ix,i_2}-in_{ix,i_2}\right)\right]=0,$$

$$m_2=1,...,TS,\quad i_2=1,...,TS;$$

3) атрибуты таблицы, используемые в теле хранимых процедур или функций, должны принадлежать хотя бы одной дочерней таблице

$$\forall px,\quad px=1,...,\left|PF\right|:\prod_{m_3}^{TS}\left[\sum_{i_3}^{TS}\left(x_{m_3,i_3}\cdot pf_{px,i_3}-pf_{px,i_3}\right)\right]=0,$$

$$m_3=1,...,TS,\quad i_3=1,...,TS;$$

4) атрибуты таблицы, используемые в работе триггеров исследуемой таблицы, должны принадлежать хотя бы одной дочерней таблице

$$\forall tx,\quad tx=1,...,\left|TG\right|:\prod_{m_4}^{TS}\left[\sum_{i_4}^{TS}\left(x_{m_4,i_4}\cdot tg_{tx,i_4}-tg_{tx,i_4}\right)\right]=0,$$

$$m_4=1,...,TS,\quad i_4=1,...,TS;$$

5) отношение количества физических блоков данных, необходимого для хранения данных рассматриваемой таблицы до применения методики, к количеству блоков, необходимому для хранения данных в дочерних таблицах, полученных после применения методики, не должно превышать заданного параметра $TSIZE$ $\left(TSIZE\in\left(0;1\right]\right)$

$$TSIZE=\frac{\sum\limits_{iq}^{nq}\left(\dfrac{RC\cdot RS\left(q_{iq},TA,DBT\right)}{DB}\right)}{\dfrac{\left(\left(\sum\limits_{j}^{TS}\left[x_{irs,j}\cdot\left(\sum\limits_{idbt}^{ndbt}dbt_{idbt}\cdot ta_j\right)\right]+RDS\left(DBT,TA,x_{irs}\right)\right)\right)\cdot RC\cdot FQ\left(q_{iq},x_{irs}\right)}{DB}}.$$

Методика нахождение оптимального разделения таблицы на дочерние для выполнения группы запросов путем многомодального распределения атрибутов таблицы по частоте их появлений в запросах

Целевая функция не линейна, а также не линейны ограничения. Переменная X – бинарная матрица размерностью $TS \times TS$. Представим переменную в виде машинного слова длиной $TS \cdot TS$. Следовательно, количество возможных комбинаций переменной определяется как $2^{TS \cdot TS}$. Исходя из этого задача обладает экспоненциальной сложностью и является *NP*-трудной [5; 6].

Для решения задачи разработана методика, основанная на многомодальном распределении атрибутов исследуемой таблицы по критерию появления их в группе запросов на чтение информации. Для получения оптимального разбиения исследуемой таблицы с числом атрибутов, равным TS, необходимо выполнить следующие действия.

1. Получить для каждого атрибута значение частоты его появления в группе запросов к базе данных. Вектор частот появлений атрибутов исследуемой таблицы в группе запросов Q, где $FTA = \left\{ fta_{ifta} \middle| ifta = \overline{1,TS} \right\}$,

$$fta_{ifta} = \sum_{iq=1}^{nq} q_{iq}\left(SFQ\right) \cdot \left(q_{iq}\left(QA\right)\right)_{ifta},$$
$$iq = 1,...,nq.$$

2. Отсортировать атрибуты по частоте их появления в группе запросов:

$$FTA' = \left\{ fta'_{ifta} \middle| ifta = \overline{1,TS},\ fta' \in FTA,\ fta'_{ifta} \le fta'_{ifta+1} \right\}.$$

3. Сформировать группы атрибутов с одинаковой частотой. Получим разбиение конечного множества FTA'. $GF = \left\{ gf_1,...,gf_{bn} \right\}$, в котором

$$gf_1 \cup ... \cup gf_{bn} = FTA',\ gf_{bi} \neq \varnothing,\ 1 \le bi \le bn,$$

где $\forall ifta,\ fta'_{ifta} \in gf_{bi}$, если $\forall \left(gf_{bi}\right)_{bj} = fta'_{ifta},\ ifta = \overline{1,TS},\ bj = \overline{1,\left|gf_{bi}\right|}$.

4. Получить разбиение множества групп атрибутов. Получим разбиение конечного множества GF. $GF' = \left\{ gf'_1,...,gf'_{gn} \right\}$, в котором

$$gf'_1 \cup ... \cup gf'_{gn} = GF,\ gf'_{gi} \neq \varnothing,\ 1 \le gi \le gn,$$

где $\forall gf_{bi},\ gf_{bi} \in gf'_{gi},\ \dfrac{\left(gi-1\right) \cdot TS}{K} \le bi \le \dfrac{gi \cdot TS}{K},\ 1 \le bi \le bn.$ K – коэффициент, характеризующий количество разбиений множества групп атрибутов, $K \in \left[1, gn\right]$.

Варьируя параметр $K \in \left[1, gn\right]$, мы можем получать решения, эффективность которых оценивается при помощи выведенной в рамках исследования целевой функции.

Практическая апробация методики

Для анализа эффективности полученной методики были выделены исходные данные и базовые множества.

1. Параметр $TS = 16$, характеризующий количество столбцов в таблице:

№ п/п	Наименование столбца	Тип данных СУБД
1	Id	bigint
2	Attr1	nchar(10)
3	Attr2	nchar(10)
4	Attr3	nchar(10)
5	Attr4	nchar(10)
6	Attr5	nchar(10)
7	Attr6	nchar(10)
8	Attr7	nchar(10)
9	Attr8	nchar(10)
10	Attr9	nchar(10)
11	Attr10	nchar(10)
12	Attr11	nchar(10)
13	Attr12	nchar(10)
14	Attr13	nchar(10)
15	Attr14	nchar(10)
16	KeyForSearch	nchar(10)

2. Множество типов данных *DBT*, которые поддерживаются конкретной выбранной СУБД MS SQL 2012. Задано вектором, характеризующим занимаемое типом данных дисковое пространство в байтах, $DBT = \{4, 8, 20\}$ (количество типов данных СУБД уменьшено для компактности). Множество типов данных СУБД MS SQL 2012 может быть представлено в виде таблицы:

№ п/п	Наименование типа данных	Занимаемое дисковое пространство, байты
1	int	4
2	bigint	8
3	nchar(10)	20

3. Набор атрибутов (столбцов таблицы) *TA*:

	DBT_1	DBT_2	DBT_3
A_1	0	1	0
A_2	0	0	1
A_3	0	0	1
A_4	0	0	1
A_5	0	0	1
A_6	0	0	1
A_7	0	0	1
A_8	0	0	1
A_9	0	0	1
A_{10}	0	0	1
A_{11}	0	0	1
A_{12}	0	0	1

Окончание таблицы

	DBT_1	DBT_2	DBT_3
A_{13}	0	0	1
A_{14}	0	0	1
A_{15}	0	0	1
A_{16}	0	0	1

4. Множество, представляющее группу запросов Q на получение информации из базы данных, состоящее из 3 элементов:

№ п/п	Количество запросов, поступивших на сервер за выбранный период времени	Бинарный вектор атрибутов, участвующих в запросе
1	10	<0,0,1,1,0,0,0,0,0,0,0,0,0,0,1,1>
2	20	<1,1,1,1,1,1,0,0,1,1,0,1,1,1,0,1>
3	5	<1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1>

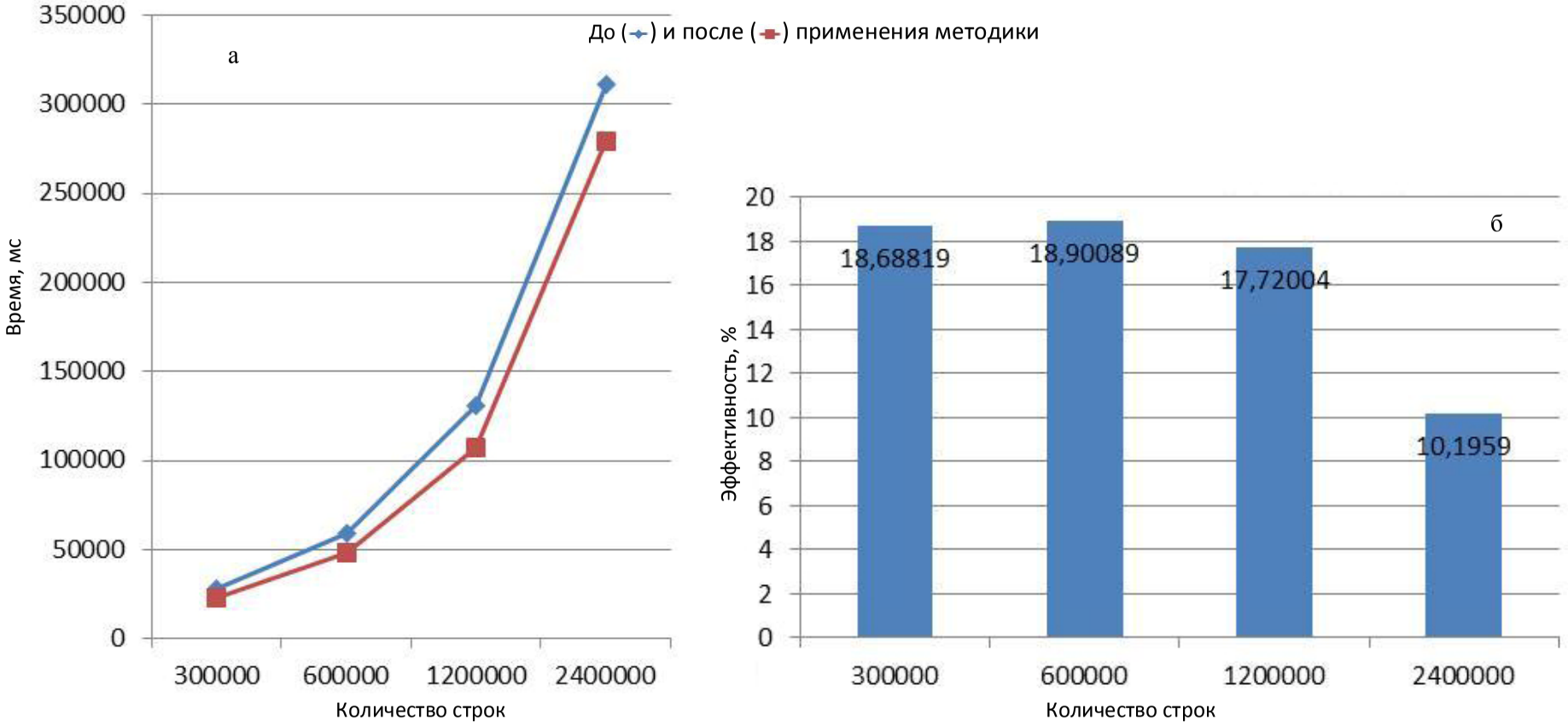
Для проведения эксперимента были исключены ограничения в виде индексов, триггеров и хранимых процедур, а также был исключен коэффициент дополнительного использования памяти. Это позволило продемонстрировать преимущества предлагаемой методики над традиционным подходом. В результате применения методики было предложено разделить исследуемую таблицу на четыре дочерних таблицы:

№	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
T1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
T3	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0
T4	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1

Полученное решение было проверено статистически на всей группе запросов. Для повышения достоверности экспериментов были выполнены все 35 запросов на чтение информации. Это позволило получить сведения о запросах, которые после применения методики стали выполняться быстрее, а также выделить подмножество медленных запросов. Результаты экспериментов, а также суммарное время выполнения группы запросов представлены в виде таблицы:

Количество строк в исследуемой таблице	Время выполнения группы запросов к исследуемой таблице, мс	Время выполнения группы запросов к таблицам после разделения на дочерние, мс
300 000	28 312	23 021
600 000	59 521	48 271
1 200 000	130 485	107 363
2 400 000	311 223	279 491

Эффективность применения методики представлена на рисунке.



Результаты применения методики в зависимости от количества строк:
а – время обработки группы запросов; *б* – эффективность, выраженная в процентах

Заключение

В результате проведенного исследования была сформулирована проблема повышения производительности информационной системы за счет реструктуризации табличных структур данных. Получено ее описание в теоретико-множественном представлении. Сформулированы целевая функция и ограничения. Предложен подход к нахождению субоптимального разбиения исследуемой табличной структуры на дочерние путем многомодального распределения атрибутов по частоте их появлений в запросах на чтение информации. Предложенная методика особенно актуальна для таблиц БД, которые используют небольшие информационные системы.

Полученные результаты могут быть использованы при проектировании отечественных СУБД. Дальнейшие исследования в этой области связаны с разработкой методик оптимальной реорганизации табличных структур данных для крупных информационных систем. Открытая апробация методики реализуется в виде веб-системы, в которой любой исследователь может ввести сведения о своей БД и получить рекомендуемое оптимальное разделение таблиц.

Список литературы

1. *Богданова А. В., Дьяченко Р. А., Бельченко И. В.* Повышение качества образовательного процесса за счет внедрения системы «Электронное расписание» в учебной организации // Политематический сетевой электронный научный журнал Кубанского государственного агарного университета. 2016. С. 873–885.

2. *Эмблер С. В., Садаладж П. Дж.* Рефакторинг баз данных: эволюционное проектирование: Пер. с англ. М.: ИД Вильямс, 2007. 672 с.

3. *Тоу Д.* Настройка SQL. Для профессионалов. СПб.: Питер, 2004. 333 с.

4. *Чигаркина Е. И.* Базы данных: Учеб. пособие. Самара: Изд-во СГАУ, 2015. 208 с.

5. *Atroshchenko V. A., BelchenkoV. E., Belchenko I. V., Dyachenko R. A.* Development and research of statistical methods and optimization algorithms of search for solutions in intelligence automated systems // International journal of pharmacy and technology. 2016. Vol. 8, no. 2. P. 14137–14149.

6. *Klir G. J.* Facets of Systems Science. Springer, 1991. 664 p.

Материал поступил в редколлегию 02.03.2018

I. V. Belchenko, **R. A. Diyachenko**

Kuban State Technological University
2 Moskovskaya Str., *Krasnodar*, *350072*, *Russian Federation*

ilur@mail.ru, *emessage@rambler.ru*

TECHNIQUES FOR IMPROVING PERFORMANCE OF THE SMALL INFORMATION SYSTEMS THROUGH OPTIMAL RESTRUCTURING DATA BASED ON MULTIMODAL DISTRIBUTIONS ATTRIBUTES

A systematic approach to increasing the productivity of small information systems is considered at the expense of optimal restructuring of tabular data structures. The authors formulated the task of optimizing the number of data blocks that are needed to query the group to read the information offered to the target function, and structural constraints. The impossibility of using crude methods of searching for the optimal solution is analyzed. The technique of multimodal attribute distribution is proposed depending on their frequency of occurrence in the query group. The experiment confirming the effectiveness of the developed methodology for small information systems.

Keywords: decision support system, optimization, data structures, databases, system analysis.

References

1. Bogdanova A. V., Diyachenko R. A., Belchenko I. V. Povyshenie kachestva obrazovatel'nogo protsessa za shchet vnedreniya sistemy «Elektronnoe raspisanie» v uchebnoj organizatsii. *Politematicheskij setevoj elektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agarnogo universiteta*, 2016, p. 873–885. (in Russ.)

2. Embler S. V., Sadaladzh P. Dzh. Refaktoring baz dannykh: evolyutsionnoe proektirovanie. Transl. from Engl. Moscow, Vilyams Publ., 2007, 672 p. (in Russ.)

3. Tou D. Nastrojka SQL. Dlya professionalov. St. Petersburg, Piter, 2004, 333 p. (in Russ.)

4. Chigarkina E. I. Bazy dannykh. Samara, SGAU Press, 2015, 208 p. (in Russ.)

5. Atroshchenko V. A., Belchenko V. E., Belchenko I. V., Diyachenko R. A. Development and research of statistical methods and optimization algorithms of search for solutions in intelligence automated systems. *International journal of pharmacy and technology*, 2016, vol. 8, no. 2, p. 14137–14149.

6. Klir G. J. Facets of Systems Science. Springer, 1991, 664 p.

For citation:

Belchenko I. V., Diyachenko R. A. Techniques for Improving Performance of the Small Information Systems through Optimal Restructuring Data Based on Multimodal Distributions Attributes. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 19–30. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-19-30