

РАСПРЕДЕЛЕНИЕ ТАБЛИЦ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

к.т.н. Г. А. Кучук, Ю. А. Акимова

Предложен подход к оптимальному размещению таблиц равного объема распределенной реляционной базы данных (РБД) на внешних запоминающих устройствах (ВЗУ) локальной вычислительной сети (ЛВС).

При проектировании сложных запросов к распределенной базе данных большой информационной емкости необходимо вложиться в заданные временные границы. В современных функциональных полных реляционных СУБД при обработке больших информационных массивов скорость обработки запроса существенно зависит от размещения на ВЗУ ЛВС таблиц, описывающих однотипные объекты [1]. При этом желательно минимизировать суммарные затраты на используемую для хранения информации внешнюю память ЛВС.

Рассмотрим запрос, формируемый на основе информации, выбираемой из M таблиц РБД объема W , причем вероятность обращения к i -ой таблице ($i = \overline{1, M}$) равна p_i , $\left(\sum_{i=1}^{M_1} p_i = 1 \right)$. Пусть T - максимально допустимое время, отведенное на обработку запроса.

Для размещения таблиц распределенной базы данных может быть предоставлено N различных типов внешней памяти со следующими характеристиками j -го типа ($j = \overline{1, N}$):

t_j - среднее время допуска к устройству типа j ;

V_j - объем стандартного информационного блока устройства типа j (причем $V_j > W \quad \forall j \in \overline{1, N}$);

n_j - максимально возможное количество блоков, предоставленных для запроса на устройствах типа j ;

f_j - средние затраты на установку одного блока типа j .

Тогда число таблиц, которые могут разместиться в одном блоке j -го типа $l_j = [V_j / W]$. Упорядочим типы ВЗУ по возрастанию времени до-

ступа, т.е. $t_j \leq t_{j+1}$, $j = \overline{1, N-1}$, а таблицы - по убыванию вероятности обращения к ним, т.е. $P_i \geq P_{i+1}$, $i = \overline{1, M-1}$.

Присвоим каждому из доступных блоков порядковый номер, независимый от типа устройства (рис.1). Общее количество доступных блоков

$K_C = \sum_{j=1}^N n_j$. Обозначим через τ_k время доступа к блоку с номером k .

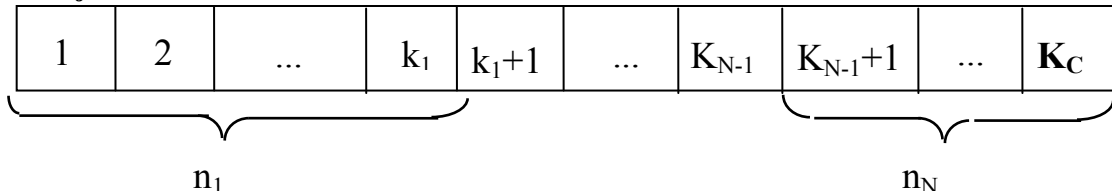


Рис. 1. Схема нумерации доступных блоков

С учетом введенной нумерации введем булевы переменные распределения таблиц по блокам внешней памяти:

$$x_{ik} = \begin{cases} 1, & \text{если } i\text{-ая таблица размещена в } k\text{-ом блоке;} \\ 0, & \text{если } i\text{-ая таблица не использует } k\text{-ый блок.} \end{cases}$$

Обозначим через y_j количество задействованных блоков типа j .

В данных обозначениях суммарные затраты на установку требуемых блоков ВЗУ составят

$$F = \sum_{j=1}^N f_j y_j. \quad (1)$$

Так как $V_j > W \quad \forall j \in \overline{1, N}$, то без ограничения общности задачи можно предположить, что каждая таблица размещается только в одном блоке,

$$\sum_{k=1}^{K_C} x_{ik} = 1, \quad i = \overline{1, M}. \quad (2)$$

Исходя из определения переменных y_j

$$y_j \leq n_j, \quad j = \overline{1, N}. \quad (3)$$

Требование о не превышении среднего времени доступа к таблицам учитывается в неравенстве

$$\sum_{i=1}^M \sum_{k=1}^{K_C} P_i \cdot x_{ik} \cdot \tau_k \leq T. \quad (4)$$

Введем на множестве $\{0, \dots, N-1\}$ функцию смещения номера - φ :
 $\varphi(0)=0$;

$$\varphi(j-1) = \sum_{k'=1}^{j-1} n_{k'}, \quad j = \overline{2, N}.$$

Функция смещения номера позволяет для устройств типа j определить порядковый номер первого доступного блока $K^{(1)}(j) = \varphi(j-1)+1$ (рис.1). Тогда число таблиц, размещенных в блоках типа j , равно

$$\left[\left(\sum_{i=1}^M \sum_{k=\varphi(j-1)+1}^{\varphi(j-1)+y_j} x_{ik} \right) \right] / I_j.$$

Поэтому требование о достаточности числа блоков типа j для размещения таблиц БД записывается как

$$\sum_{i=1}^M \sum_{k=\varphi(j-1)+1}^{\varphi(j-1)+y_j} x_{ik} \leq I_j \cdot y_j \quad \forall j \in \overline{1, N} \quad (5)$$

Кроме того, $x_{ik} \in \{0, 1\}$, $y_j \in \{0, 1, \dots, k_j\}$. (6)

Требование минимизации суммарных затрат приводит к линейной задаче целочисленного программирования с целевой функцией

$$F = \sum_{j=1}^N f_j y_j \rightarrow \min \quad (7)$$

и ограничениями (2) - (6).

Наличие переменной y_j в пределах суммирования ограничения (5) не позволяет решить задачу (7) стандартными методами. При небольших N и M возможно решение методом полного перебора множества допустимых решений. При проектировании реляционных баз данных, содержащих большое количество таблиц и элементарных динамических запросов, необходимо использовать методы частичного перебора, учитывающие специфику ограничений (2) - (5).

Литература

1. Жожикашвили В. А., Вишневский В. М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. - М.: Радио и связь, 1988. - 189с.