

РАЦИОНАЛЬНОЕ РАЗМЕЩЕНИЕ ФАЙЛОВ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ В ВЫЧИСЛИТЕЛЬНОЙ СЕТИ С ПРОИЗВОЛЬНОЙ ТОПОЛОГИЕЙ

Ладыженский Ю.В., Бельков Д.В.

Кафедра ПМИИ ДонГТУ

ladyzhen@pmi.donetsk.ua

Abstract

Ladyzhensky Y.V., Belkov D.V. Rational database files location for the network with arbitrary topology. Computer networks need rational locations of database files. Mathematical models and algorithms to determine of number of file copies and to locate them in a network with arbitrary topology are proposed. Results of experiments are presented.

Введение

Задача оптимального размещения файлов распределенной базы данных (РБД) по узлам возникает при проектировании и при эксплуатации вычислительных сетей. Особую актуальность она приобретает в связи с развитием сети INTERNET, в государственных и корпоративных сетях управления, в локальных сетях для автоматизированного проектирования.

При создании РБД необходимы исследования организации информации в вычислительных системах с помощью математических моделей. Задачи оптимального распределения файлов РБД, где в качестве критерия оптимальности принимается общее среднее время ответа на запрос (Т), сводятся к задаче целочисленного нелинейного программирования с линейными ограничениями. Недостаток этого подхода в том, что необходимо аналитически выразить Т как функцию распределения файлов, для чего требуется значительное число исходных данных, знание которых не всегда доступно заранее.

В данной статье для уменьшения времени ответа на запрос оптимизируется интенсивность информационного трафика. При функционировании сети в каждом узле образуются 2 типа запросов: сетевой запрос, для обработки которого необходим файл, не содержащийся в том узле, где возник запрос; локальный запрос, для обработки которого необходим файл, содержащийся в том узле, где возник запрос.

Критерием оптимальности размещения файлов является суммарный поток сетевых запросов, инициированных в узлах в единицу времени. Чем меньше суммарный поток таких запросов, тем меньше объем данных, необходимый для пересылки, а значит, меньше среднее время реакции системы на запросы. Очевидно, чем больше суммарный поток локальных запросов, тем меньше среднее время реакции системы на запросы. Поэтому суммарный поток локальных запросов также служит критерием оптимальности.

В работе [1] предложена утилита для размещения файлов по компьютерам сети. Утилита решает задачу:

$$\sum_i \sum_j F_{ij} X_{ij} \rightarrow \max; \sum_i X_{ij} M_i \leq B_j, \text{ где } X_{ij} = 1, \text{ если файл } i \text{ размещен в узле } j, \text{ иначе}$$

$X_{ij} = 0$; M_i - размер файла i ; B_j - объем узла j ; F_{ij} - частота обращения к файлу i из узла j . Для решения задачи разработана эвристическая процедура, основанная на идеях алгоритма решения задачи об упаковке в контейнеры.

В первой части данной статьи сформулирована задача рационального размещения файлов с максимизацией количества локальных запросов. Для ее решения используется модификация алгоритма, предложенного в [2]. Все файлы располагаются в локальных базах узлов так, что имеется только одна копия каждого файла.

Во второй части сформулирована задача и предложен алгоритм размещения по узлам файлов с копиями, вычисляется оптимальное количество копий файлов.

Для экспериментов использовалась ПЭВМ Pentium - 166, программы составлены на языке Turbo Pascal 7.1, операционная система MSDOS 7.1. Задачи решены с помощью описанных в статье методов. Один шаг алгоритма означает одну операцию сравнения; T_r - время работы алгоритма. Топология сети считалась произвольной. Результаты решения задач приведены в таблицах 1 и 2.

1. Размещение файлов с максимизацией потока локальных запросов

1.1. Постановка задачи

Пусть m - количество файлов РБД; n - количество узлов сети; файл i находится в узле j ; λ_j - средняя интенсивность запросов, инициированных в узле j ; P_{ij} - вероятность того, что для обслуживания запроса, инициированного в узле j , необходим файл i ; X - матрица размещения файлов: $X_{ij} = 1$, если файл i находится в узле j , иначе $X_{ij} = 0$; λ_{jk} - средняя интенсивность поступления запросов типа k во входную очередь узла j , $k=1,2$; W_{jk} - среднее время обработки запроса типа k в узле j ; $i=1...m$; $j=1...n$.

Целевая функция:

$$\sum_{i=1}^m \sum_{j=1}^n \lambda_j P_{ij} X_{ij} \rightarrow \max \quad (1.1)$$

Ограничения:

$$\sum_{j=1}^n X_{ij} = 1; \sum_{j=1}^n \lambda_{j1} W_{j1} + \lambda_{j2} W_{j2} \leq 1 \quad (1.2)$$

Необходимо найти матрицу X , обеспечивающую максимум целевой функции (1.1) при ограничениях (1.2).

1.2. Алгоритм размещения файлов

1. Формируем матрицу C с элементами: $C_{ij} = \lambda_j P_{ij}$;
2. Задаем нулевые значения матрице размещения файлов X и матрице запретов размещения E ;
3. Изменяем матрицу C : если $\lambda_{j1} W_{j1} + \lambda_{j2} W_{j2} > 1$, то $C_{ij} := 0$, $i=1...m$, $j=1...n$;
4. Полагаем $\max = C_{i_1}$ и $p=1$.
5. Среди не запрещенных узлов ($E_{ij} \neq 1$) для каждого файла i находим $C_{ip} = \max C_{ij}$, $j=1...n$;
6. Если $\lambda_{p1} W_{p1} + \lambda_{p2} W_{p2} \leq 1$, то файл i размещаем в узле p : $X_{ip} := 1$, иначе:

- а) запрещаем узел $p : E_p := 1$;
 - б) среди всех узлов находим первый не запрещенный узел s , для которого $\lambda_{s1}W_{s1} + \lambda_{s2}W_{s2} \leq 1$;
 - в) полагаем $p=s$ и $\max = C_p$;
 - г) переходим к пункту б.
7. Используя сформированную матрицу X , вычисляем функцию цели по формуле (1.1).

На этапе бб может оказаться, что все узлы запрещены для размещения файла i : $X_{ij} = 0$ для всех $j=1...n$, т.е. алгоритм не может решить задачу при заданных начальных значениях. В этом случае необходимо изменить параметры узлов, добившись уменьшения W_{jk} и повторить вычисления с шага 1.

1.3. Пример решения задачи

Рассмотрим задачу: $m = 8$; $n = 3$;

Матрица P :	Вектор λ :	Матрица W :																																	
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0.6</td><td>0.1</td><td>0</td></tr> <tr><td>0.3</td><td>0.1</td><td>0</td></tr> <tr><td>0.1</td><td>0.05</td><td>0.5</td></tr> <tr><td>0</td><td>0</td><td>0.2</td></tr> <tr><td>0</td><td>0.05</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0.1</td></tr> <tr><td>0</td><td>0.7</td><td>0.1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0.6	0.1	0	0.3	0.1	0	0.1	0.05	0.5	0	0	0.2	0	0.05	0	0	0	0.1	0	0.7	0.1	0	0	0	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>18</td><td>2</td><td>2</td></tr> </table>	18	2	2	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0.1</td><td>0.05</td></tr> <tr><td>0.1</td><td>0.05</td></tr> <tr><td>0.1</td><td>0.05</td></tr> </table>	0.1	0.05	0.1	0.05	0.1	0.05
0.6	0.1	0																																	
0.3	0.1	0																																	
0.1	0.05	0.5																																	
0	0	0.2																																	
0	0.05	0																																	
0	0	0.1																																	
0	0.7	0.1																																	
0	0	0																																	
18	2	2																																	
0.1	0.05																																		
0.1	0.05																																		
0.1	0.05																																		

Решение этой задачи получено на 12 шаге предлагаемого алгоритма за 0.09 секунды. Результат приведен в таблице 1.

Таблица 1.

<i>Размещение файлов</i>			<i>Кол. Шагов</i>	<i>Тр, сек.</i>
Узел 1	Узел 2	Узел 3		
F1;F2;F3	F5;F7	F4;F6;F8	12	0.09

2. Размещение копий файлов

2.1. Постановка задачи

Пусть m - количество файлов; n - количество узлов сети; B_j - объем узла j ; V_i - объем файла i ; Y_i - число копий файла i ; F_{ij} - частота запросов к файлу i из узла j . B_j - объем узла j . Пусть $X_{ij}=1$, если файл i размещен в узле j , иначе $X_{ij}=0$.

Целевая функция:
$$\sum_{i=1}^m \sum_{j=1}^n F_{ij} X_{ij} V_i \rightarrow \max, i=1...m \tag{2.1}$$

Ограничения:

$$X_{ij} \in \{0; 1\}, \sum_{j=1}^n X_{ij} = Y_i, i=1 \dots m \quad (2.2)$$

$$\sum_{i=1}^m V_i X_{ij} \leq B_j, j=1 \dots n \quad (2.3)$$

Необходимо найти матрицу размещения файлов X , обеспечивающую максимум целевой функции (2.1) при ограничениях (2.2), (2.3). Для решения этой задачи нужно сначала определить оптимальное количество копий файлов $Y_i, i=1 \dots m$.

2.2. Определение оптимального количества копий файлов

Процесс чтения вызывает передачу справочных запросов и читаемых файлов, поэтому целесообразно размещение файлов в каждом узле сети. Интенсивность передачи справочной информации в этом случае равна 0. При уменьшении числа копий она возрастает и достигает максимума при $Y_i=1, i=1 \dots n$. Однако, если все файлы дублировать, то любое обновление данных придется выполнять во всех узлах сети. С увеличением числа копий интенсивность трафика корректирующих запросов возрастает, достигая максимума при $Y_i=n, i=1 \dots n$.

Пусть частота справочных запросов узла j ко всем копиям файла i равна R_{ij}/Y_i , а частота корректирующих запросов узла j ко всем копиям файла i есть $W_{ij}Y_i$. Очевидно, чтобы найти оптимальное количество копий каждого файла, нужно решить уравнение $R_{ij}/Y_i = W_{ij}Y_i$. Достаточное условие наличия Y_i копий файла i определяет отношение $W_{ij}/R_{ij} = K_{ij}$. Целесообразно иметь в сети Y_i копий файла i , если копии файлов корректируются редко: $K_{ij} \leq 1$ для всех i и j . Таким образом:

$$Y_i = \sqrt{R_{ij}/W_{ij}}, \text{ если } K_{ij} \leq 1, \text{ иначе } Y_i=1; i=1 \dots m, j=1 \dots n \quad (2.4)$$

2.3. Алгоритм размещения копий файлов

1. Формируем вектор Y по формуле (2.4) и матрицу C с элементами $C_{ij} = F_j V_i$;
2. Задаем нулевые значения матрице размещения файлов X и матрице запретов размещения E ;
3. Задаем нулевые значения вектору заполнения узлов T ;
4. Изменяем матрицу C : если $V_i > B_j$, то $C_{ij} := 0$;
5. Полагаем $\max = C_{11}$ и $p=1$.
6. Среди не запрещенных узлов ($E_{ij} \neq 1$) для каждого файла i находим $C_{ip} = \max C_{ij}, j=1 \dots n$;
7. Если файл i помещается в узел p , т.е. $T_p + V_i \leq B_p$, то размещаем его в узле p : $X_{ip} := 1$; $T_p := T_p + V_i$, а также изменяем счетчик копий файлов: $Y_i := Y_i - 1$.
Если файл i не помещается в узел p или $Y_i \geq 1$, то:
 - а) запрещаем узел p : $E_{ip} := 1$;
 - б) среди всех узлов находим первый, не запрещенный узел s , в который помещается файл i ;
 - в) полагаем $p=s$ и $\max = C_{ip}$;
 - г) переходим к пункту 6.

8. Используя сформированную матрицу X, вычисляем функцию цели по формуле (1).

На этапе 76 может оказаться, что все узлы запрещены для размещения очередной копии файла i: $\sum_{j=1}^n X_{ij} < Y_i$, т.е. алгоритм не может решить задачу при заданных начальных условиях. В этом случае необходимо увеличить объем узлов и повторить вычисления с шага 1.

2.4. Пример решения задачи

Рассмотрим задачу : m= 8; n= 3;

Матрица F :	Вектор Y :	Вектор Y :	Вектор B :																																											
<table border="1"> <tr><td>1</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>5</td><td>3</td></tr> <tr><td>3</td><td>1</td><td>4</td></tr> <tr><td>4</td><td>2</td><td>5</td></tr> <tr><td>5</td><td>3</td><td>1</td></tr> <tr><td>1</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>5</td><td>3</td></tr> <tr><td>3</td><td>1</td><td>4</td></tr> </table>	1	4	2	2	5	3	3	1	4	4	2	5	5	3	1	1	4	2	2	5	3	3	1	4	<table border="1"> <tr><td>8</td></tr> <tr><td>7</td></tr> <tr><td>6</td></tr> <tr><td>5</td></tr> <tr><td>4</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> </table>	8	7	6	5	4	3	2	1	<table border="1"> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> </table>	1	2	1	2	1	3	2	3	<table border="1"> <tr><td>13</td><td>21</td><td>24</td></tr> </table>	13	21	24
1	4	2																																												
2	5	3																																												
3	1	4																																												
4	2	5																																												
5	3	1																																												
1	4	2																																												
2	5	3																																												
3	1	4																																												
8																																														
7																																														
6																																														
5																																														
4																																														
3																																														
2																																														
1																																														
1																																														
2																																														
1																																														
2																																														
1																																														
3																																														
2																																														
3																																														
13	21	24																																												

При решении этой задачи оптимальное распределение копий файлов получено на 142 шаге за 1.07 секунды. Результат приведен в таблице 2.

Таблица 2.

Размещение файлов			Кол. Шагов	Тр,сек.
Узел 1	Узел 2	Узел 3		
F4;F5; F6;F8	F1;F2;F6; F7;F8	F2;F3;F4;F6; F7;F8	142	1.07

3. Исследование быстродействия алгоритмов

Были решены три задачи размещения m файлов по n узлам для n=3 и m=8; 16; 32. Обозначим: T1_p - время работы алгоритма решения задачи (1.1-1.2); T2_p - время работы алгоритма решения задачи (2.1-2.3). График зависимости T1_p(m) показан на рисунке 1, T2_p(m) - на рисунке 2. Время решения с увеличением m растет значительно медленнее, чем экспонента вида n^m = 3^m.

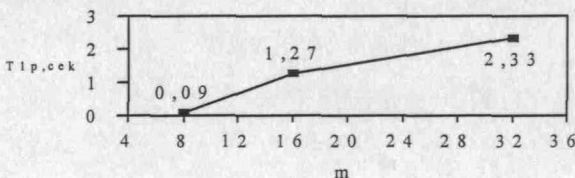


Рис. 1.

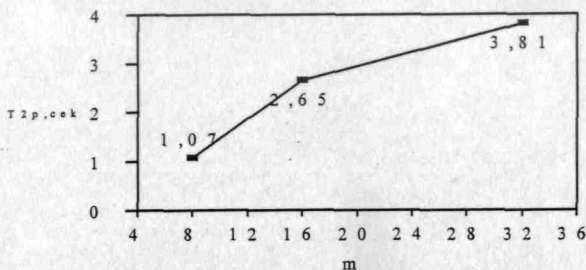


Рис.2.

Заключення

Размещение данных по узлам сети - одна из ключевых проблем при создании и эксплуатации такой системы. Эта задача является NP- трудной, принадлежит к классу задач, для которых не известен точный алгоритм с временем решения, растущим полиномиально с размерностью задачи.

Для известных точных алгоритмов время решения этих задач растет экспоненциально с размерностью задачи, т.е. если m представляет размерность задачи, то время решения растет как n^m , где $n > 1$. Таким образом, эти алгоритмы нельзя применять для работы в реальном времени. Поэтому актуальной является разработка алгоритмов, время решения которых растет полиномиально: m^n , где m - размерность задачи.

В статье предложены эвристические полиномиальные алгоритмы решения задачи размещения данных. Они имеют важное практическое значение для проектировщиков вычислительных сетей.

Литература

1. Гукасян Н.А. Метод и утилита для проектирования структуры локальной сети. //Управляющие системы и машины, 1994, N 1/2.- С. 33-35.
2. Ладженский Ю.В., Бельков Д.В. Алгоритм рационального размещения файлов распределенной базы данных по узлам вычислительной сети с кольцевой топологией. В кн. Информатика, кибернетика и вычислительная техника (ИКВТ-99). Сборник трудов ДонГТУ, Выпуск 2. Донецк: ДонГТУ, 1999, С. 272-275.