РАЗМЕЩЕНИЕ ДАННЫХ В ПАМЯТИ УЗЛА КОМПЬЮТЕРНОЙ СЕТИ

Ладыженский Ю.В., Бельков Д.В.
Кафедра ПМиИ ДонГТУ
ladyzhen@pmi.donetsk.ua

Abstract

Ladyzhensky Y.V., Belkov D.V. Database files location to knot memory for network. *Computer networks need rational locations of database files in knot memory. Mathematical models and algorithms to locate files in knot memory for network are proposed.*

Введение

Распределение информации влияет на важнейшие параметры распределенной системы: объемы хранимых и обновляемых данных, интенсивность потоков данных, быстродействие и надежность. Для рационального размещения данных системы необходимо оптимальное размещение файлов по узлам и в каждом узле сети.

Эти задачи являются NP- трудными , принадлежат к классу задач, для которых не известен точный алгоритм, со 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 $O(n^m)$, где $n>1$. Таким образом, эти алгоритмы нельзя применять для работы в реальном времени. Поэтому актуальной является разработка приближенных алгоритмов, время решения которых растет полиномиально как $O(m^n)$, где m - размерность задачи.

Для оптимального распределения информации в памяти ЭВМ в данной статье сформулированы задачи линейного программирования с булевскими переменными. Задачи решены с помощью описанных в статье алгоритмов.

1. Размещение файлов по запоминающим устройствам

1.1. Постановка задачи

В узле сети предполагается следующая структура памяти: кэш, ОЗУ, диск. Пусть m - количество файлов, размещаемых в узле; n - количество запоминающих устройств (ЗУ), n=3: j=1 - кэш-память, j=2 - буферный пул ОЗУ, j=n - диск. Пусть V_i - объем файла i; F_i - частота запросов к файлу i ; B_j - объем ЗУ j; W_j - быстродействие устройства j. Пусть X_{ij}=1, если файл i размещен в ЗУ j, иначе X_{ij}=0. Запоминающие устройства упорядочены по убыванию быстродействия и по возрастанию объема. В таком случае, для размещения файлов с высокой частотой запросов в ЗУ с наибольшим быстродействием, необходима максимизация $F_iV_iW_j$ для всех i и j. Задача линейного программирования с булевскими переменными имеет вид:

$$\text{Целевая функция: } \sum_{j=1}^{n}\sum_{i=1}^{m} F_iV_iW_jX_{ij} / B_j \to \max \qquad (1.1)$$

Ограничения:

$$X_{ij}=\{0;1\}, \sum_{j=1}^{n} X_{ij}=1, \text{ i=1...m} \tag{1.2}$$

$$\sum_{i=1}^{m} V_i X_{ij} \le B_j, \text{ j=1...n} \tag{1.3}$$

Необходимо найти матрицу размещения файлов X, обеспечивающую максимум целевой функции (1.1) при ограничениях (1.2),(1.3). Алгоритм решения задачи предложен в [1].

1.2. Алгоритм решения задачи

1. Формируем матрицу C с элементами $C_{ij} = F_i V_i W_j / B_j$;
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 $= C_{i1}$ и p=1.
6. Среди не запрещенных узлов ($E_{ij} \neq 1$) для каждого файла i находим $C_{ip} = \max C_{ij}$,j=1...n;

$C_{ip} = \max C_{ij}$,j=1...n;

7. Если файл i помещается в узел p, т.е. $T_p + V_i \le B_p$, то размещаем его в узле p: $X_{ip} := 1$; $T_p := T_p + V_i$.

Если файл i не помещается в узел p, 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 (1.1).

На этапе 7б может оказаться, что все узлы запрещены для размещения файла i: т.е. алгоритм не может решить задачу при заданных начальных условиях. В этом случае необходимо увеличить объем устройств и повторить вычисления с шага 1.

2. Размещение файла на диске

Время доступа к НМД является одним из факторов, влияющих на эффективность работы вычислительной системы. Это время тратится на перевод магнитных головок в процессе выполнения запросов. Его можно уменьшить за счет согласования расположения информации на накопителе с последовательностью обращений. Задача заключается в таком размещении информации, при котором среднее время доступа будет минимальным. Для уменьшения времени доступа целесообразно группировать информацию по цилиндрам таким образом, чтобы обеспечить высокую локальность обращений - большую вероятность последовательных обращений к одному и тому же цилиндру. Отличие частот запросов к соседним записям должно быть минимальным.

Пусть m - число записей файла; n - число цилиндров на диске; V_i - длина записи i; F_i - частота запросов к записи i; B_j - объем цилиндра j в записях. Пусть X_{ij}=1, если

запись i размещена на цилиндре j, иначе X_{ij}=0. Задача линейного программирования с булевскими переменными имеет вид:

Целевая функция: $$\sum_{i=1}^{m}\sum_{j=1}^{n}(F_iV_iX_{i,j} - F_{i-1}V_{i-1}X_{i-1,j})/B_j \to \min \quad (2.1)$$

Ограничения:

$$F_0 = 0,\ V_0 = 0,\ X_{0,j} = 0,\ X_{ij} = \{0;1\},\ \sum_{j=1}^{n} X_{ij} = 1,\ \text{i=1...m} \quad (2.2)$$

$$\sum_{i=1}^{m} V_iX_{ij} \le B_j,\ \text{j=1...n} \quad (2.3)$$

Необходимо найти матрицу размещения записей X, обеспечивающую минимум целевой функции (2.1) при ограничениях (2.2),(2.3). Алгоритм решения задачи предложен в [2]. В нем, в отличие от алгоритма, описанного в разделе 1.2. формируется матрица C с элементами $C_{ij} = (F_iV_i - F_{i-1}V_{i-1})/B_j$ и вычисляется минимум, а не максимум целевой функции.

3. Результаты экспериментов

Для исследования предложенных алгоритмов проведены серии экспериментов на ПЭВМ Pentium - 166. В первой и третьей сериях программы составлены на языке Turbo Pascal 7.1, операционная система MSDOS 7.1. Во второй и четвертой сериях программы составлены на Delphi 3.01, операционная система Windows 98.

Были решены следующие задачи:

1) 10 задач размещения m файлов, m=10,20,…,100, по n ЗУ при n=3. Элементы матрицы F_iW_j формировались функцией random(100);
2) 10 задач размещения m записей, m=20,30,…,110, по n цилиндрам для n =10. Элементы матрицы C_{ij} формировались функцией random(100).

В каждом эксперименте вектор V сформирован по формулам: $V_1 = m$; $V_i = V_1 - (i-1)$, где i=1…m. $B_1 = \sum_{i=1}^{m} V_i / n$; $B_j = B_1 + (j-1)$, j=1…n. Таким образом, размещаемые элементы упорядочены по убыванию размеров: $V_i \ge V_{i+1}$, i=1…m-1, устройства размещения упорядочены по возрастанию объема: $B_j \le B_{j+1}$, j=1…n-1.

В первой серии экспериментов матрица F_iW_j задачи размещения файлов и матрица C_{ij} задачи размещения записей наращиваются с увеличением m. Таким образом, матрица размерности (m,n) является частью матрицы размерности (m+10.n). Во второй серии экспериментов эти матрицы не наращиваются.

Обозначим: k_i - число невыполненных неравенств $\sum_{i=1}^{m} V_iX_{i,j} \le B_j$ при размещении элемента i, $0 \le k_i \le n$. Если $k_i = 0$, то элемент i обеспечивает максимальное слагаемое в целевой функции, если $k_i = n-1$, то обеспечивает минимальное слагаемое. В случае $k_i = n$ алгоритм не может разместить элемент i. Обозначим: K_c -количество операций сравнения, $K_c = n\sum_{i=1}^{m}(1+k_i)$. Обозначим: T_p - время работы алгоритма, T_c - время выполнения операции сравнения. Значения T_p и

K_c определялись для каждой задачи экспериментально, $T_c = T_p / K_c$. В первой серии экспериментов при размещении m файлов по n устройствам получены результаты, приведенные в таблице 1, $T_c \approx 0{,}5 \cdot 10^{-3}$ секунды. В случае размещения m записей по n цилиндрам получены результаты, приведенные в таблице 2, $T_c \approx 0{,}15 \cdot 10^{-3}$ секунды. Во второй серии экспериментов при размещении m файлов по n устройствам получены результаты, приведенные в таблице 3, $T_c \approx 3 \cdot 10^{-6}$ секунды. В случае размещения m записей по n цилиндрам получены результаты, приведенные в таблице 4, $T_c \approx 2 \cdot 10^{-6}$ секунды.

При неограниченных по объему устройствах, задачи решаются строго точно за время $T_{min} = nmT_c$. Временная сложность алгоритма, если устройства ограничены по объему - $T_{max} = mn^2 T_c$. Алгоритм обеспечивает приближенное решение для таких задач. Графики зависимостей $T_p(m), T_{min}(m), T_{max}(m)$ показаны на рисунках 1-4. Время решения с увеличением m растет значительно медленнее, чем экспонента n^m.

Таблица 1.

№	Тр мсек.	Кс	Тс мсек.
1	21,4	42	0,51
2	41,2	81	0,51
3	60,9	117	0,52
4	80,6	165	0,49
5	100,6	192	0,52
6	120,3	231	0,52
7	140,6	261	0,54
8	159,9	297	0,54
9	180,1	324	0,56
10	200	357	0,56

Таблица 2.

№	Тр мсек.	Кс	Тс мсек.
1	44	300	0,15
2	64,8	450	0,14
3	86,2	600	0,14
4	107,1	770	0,14
5	128,5	930	0,14
6	149,9	1010	0,15
7	171,4	1180	0,15
8	192,3	1300	0,15
9	213,1	1390	0,15
10	235,1	1550	0,15

Таблица 3.

№	Тр мсек.	Кс	Тс мсек.
1	0,11	36	0,003
2	0,22	90	0,0024
3	0,27	126	0,0021
4	0,44	165	0,0027
5	0,55	195	0,0028
6	0,66	243	0,0027
7	0,72	246	0,0029
8	0,82	303	0,0027
9	0,94	387	0,0024
10	0,99	333	0,003

Таблица 4.

№	Тр мсек.	Кс	Тс мсек.
1	0,61	310	0,002
2	0,99	680	0,001
3	1,26	630	0,002
4	1,59	870	0,002
5	1,92	1050	0,002
6	2,2	1060	0,002
7	2,53	1360	0,002
8	2,85	1510	0,002
9	3,18	1470	0,002
10	3,52	2380	0,001

В третьей и четвертой сериях экспериментов исследовалась точность решений, полученных алгоритмом в задаче размещения файлов по устройствам. Были решены 10 задач для 8 файлов и 3 - х устройств методом полного перебора и описанным в разделе 1.2. алгоритмом. В каждой задаче элементы матрицы $F_i W_j$ формировались функцией random(100). Обозначим: Al - значение целевой функции, полученное алгоритмом, Op -

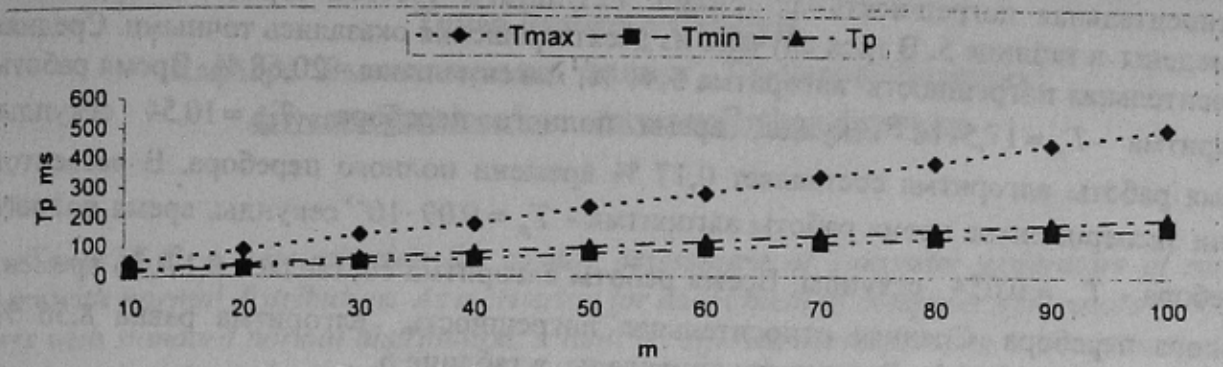

Рисунок 1 - Эксперимент 1. Размещение файлов по ЗУ.

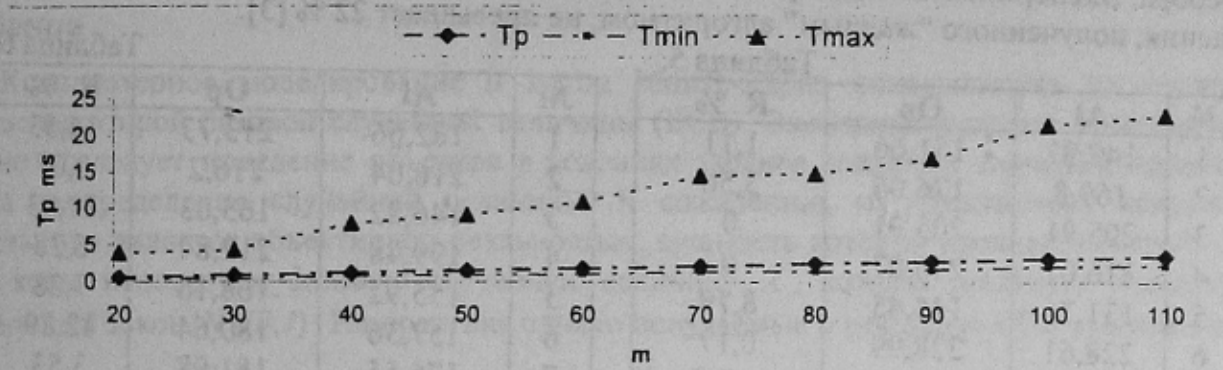

Рисунок 2 - Эксперимент 4. Размещение записей по цилиндрам.

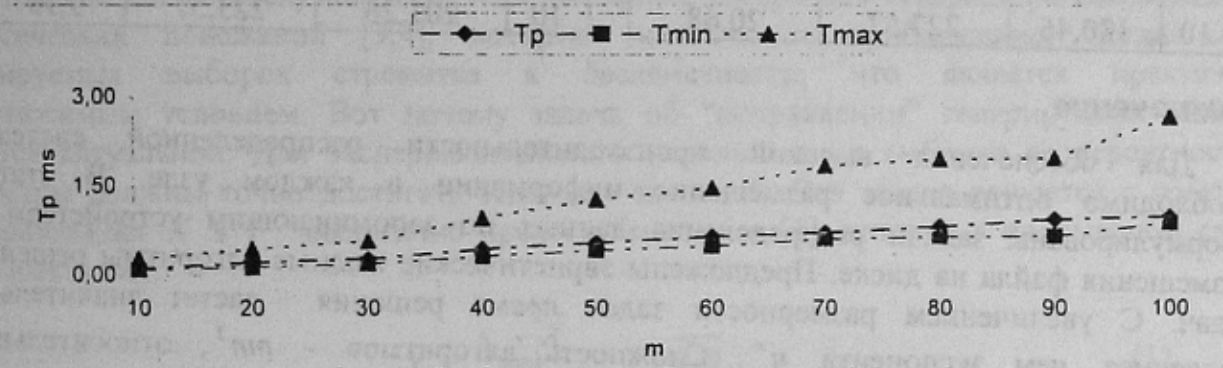

Рисунок 3 - Эксперимент 2. Размещение файлов по ЗУ.

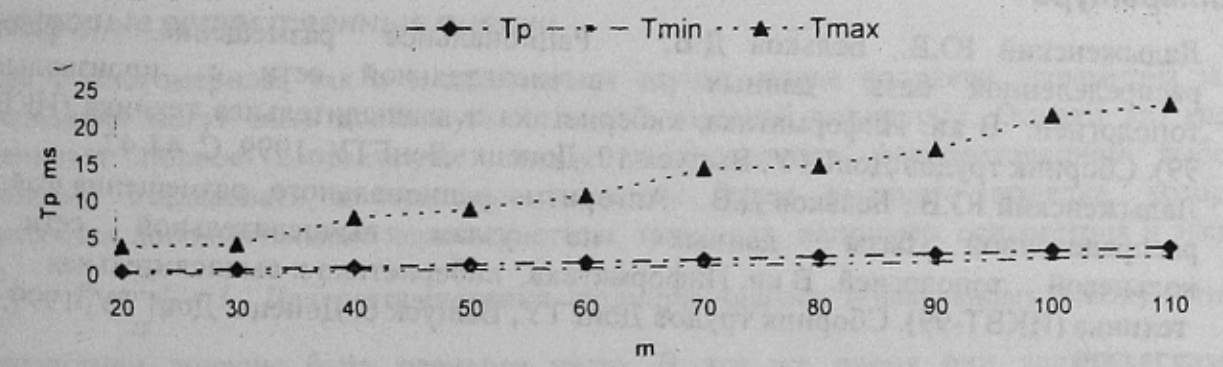

Рисунок 4 - Эксперимент 4. Размещение записей по цилиндрам.

значение целевой функции при строго оптимальном решении, $R = |A_i - O_p| * 100\% / O_p$ - относительная погрешность алгоритма. Результаты третьей серии экспериментов приведены в таблице 5. В трех случаях из десяти решения оказались точными. Средняя относительная погрешность алгоритма 5,41 %, максимальная - 20,68 %. Время работы алгоритма - $T_p = 17{,}5 \cdot 10^{-3}$ секунды, время полного перебора - $T_{op} = 10{,}54$ секунды. Время работы алгоритма составляет 0,17 % времени полного перебора. В четвертой серии экспериментов время работы алгоритма - $T_p = 0{,}09 \cdot 10^{-3}$ секунды, время полного перебора - $T_{op} = 0{,}054$ секунды. Время работы алгоритма составляет 0,17 % времени полного перебора. Средняя относительная погрешность алгоритма равна 8,56 %, максимальная - 17,65 %. Результаты приведены в таблице 6.

В этих экспериментах алгоритм решил задачи в 600 раз быстрее полного перебора. Эксперименты подтверждают, что теоретическая относительная погрешность решения, полученного "жадным" алгоритмом, не превышает 22 % [3].

Таблица 5.

№	Al	Op	R %
1	149,97	151,66	1,11
2	169,8	176,06	3,56
3	205,91	205,91	0
4	216,07	216,07	0
5	131,76	144,45	8,79
6	228,61	228,99	0,17
7	214,76	214,76	0
8	164,3	190,02	13,54
9	165,96	177,01	6,24
10	180,46	227,52	20,68

Таблица 6.

№	Al	Op	R %
1	182,66	213,73	14,53
2	216,04	216,2	0,07
3	146,82	165,03	11,03
4	199,48	212,84	6,28
5	155,92	168,16	7,28
6	157,36	180,64	12,89
7	175,55	181,98	3,53
8	140,1	170,12	17,65
9	147,04	150,64	2,39
10	201,26	223,57	9,98

Заключение

Для обеспечения высокой производительности распределенной системы необходимо оптимальное размещение информации в каждом узле. В статье сформулированы задачи распределения данных по запоминающим устройствам и размещения файла на диске. Предложены эвристические жадные алгоритмы решения задач. С увеличением размерности задач время решения растет значительно медленнее, чем экспонента n^m. Сложность алгоритмов - mn^2, относительная погрешность не превышает 22 %.

Литература

1. Ладыженский Ю.В., Бельков Д.В. Рациональное размещение файлов распределенной базы данных в вычислительной сети с произвольной топологией. В кн. Информатика, кибернетика и вычислительная техника (ИКВТ-99). Сборник трудов ДонГТУ, Выпуск 10. Донецк: ДонГТУ, 1999, С. 44-47.
2. Ладыженский 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 6. Донецк: ДонГТУ, 1999, С. 272-275.
3. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. - Москва: 1982. - 416 с.

Поступила в редакційну колегію 1.03.2000 р.