

Д.В. Бельков

**АЛГОРИТМ РАЗМЕЩЕНИЯ ФАЙЛОВ ПО УЗЛАМ
ВЫЧИСЛИТЕЛЬНОЙ СЕТИ РАСПРЕДЕЛЕННОЙ САПР**

Сформулирована задача размещения файлов распределенной САПР по узлам вычислительной сети. В качестве оптимизируемого критерия размещения файлов предложена максимизация суммарного времени использования файлов узлов. Ограничениями в задаче служат объемы памяти и коэффициенты загрузки узлов. Предложен эвристический алгоритм решения задачи. Приведены результаты исследования алгоритма. Табл. 1, ист. 2.

Задача оптимального размещения файлов распределенной системы по узлам вычислительной сети 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 [1]. Наиболее подходящим построением САПР может быть объектно-ориентированная машина распределенной архитектуры [2]. Частью машины САПР является собственная распределенная операционная система. В эту систему входит оптимизатор, который производит самоадаптацию размещения файлов в зависимости от интенсивности их использования. В статье предлагается возможный алгоритм работы оптимизатора.

Обозначим: F_{ij} - количество запросов к файлу i из узла j в единицу времени; $X_{ij}=1$, если файл i расположен в узле j , иначе $X_{ij}=0$; V_i - объем файла i ; B_j - объем узла j , $i=1\dots m$, $j=1\dots n$. Значения F_{ij} случайны и зависят от времени t , остальные величины постоянны. Обозначим: T_j - время использования файлов узла j ; θ_j - допустимое время выполнения запроса к файлам узла j ; ρ_j - коэффициент загрузки узла j : $\rho_j = \sum_{i=1}^m F_{i,j} \cdot \theta_j$. Если $0 < \rho_j < 1$, то в узле j нет очереди запросов к файлам. Если $\rho_j > 1$, то в узле j возникает очередь запросов к файлам. Поэтому в случае $\rho_j > 1$ время реакции узла на запрос больше, чем в случае $0 < \rho_j < 1$. Для эффективной работы системы необходимо выполнение условий: $\rho_j \rightarrow \max$; $\rho_j < 1$, $T_j \rightarrow \max$, $j=1\dots n$. Поэтому файлы должны быть размещены по узлам с целью максимизации функции (1) при ограничениях (2)-(4).

Целевая функция:

$$T = \sum_{i=1}^m \sum_{j=1}^n \rho_j X_{ij} T_j V_i / B_j = \sum_{i=1}^m \sum_{j=1}^n C_{i,j} X_{i,j} \rightarrow \max \quad (1)$$

Ограничения:

$$X_{ij} \in \{0,1\}, \sum_{j=1}^n X_{ij} = 1, i=1\dots m \quad (2)$$

$$\sum_{i=1}^m V_i X_{ij} \leq B_j, j=1...n \quad (3)$$

$$\rho_j < 1, j=1...n \quad (4)$$

В задаче (1)-(4) максимизируется суммарное время использования файлов узлов. В ней необходимо найти матрицу размещений файлов X . Каждый файл размещается только в один узел. Задача относится к классу NP-трудных, так как содержит в качестве частного случая многомерную задачу о рюкзаке.

Эвристический алгоритм решения задачи (1)-(4) размещает файлы по принципу "в первый подходящий узел в порядке убывания размеров файлов" (FFD). Перед применением алгоритма файлы нужно упорядочить по убыванию их размеров, а узлы – по возрастанию объемов. Работа алгоритма состоит из двух этапов. На первом этапе алгоритм находит для файла объемом V те узлы, в которые файл помещается по размеру. На втором этапе, среди найденных узлов определяется узел с наибольшим значением $C_j = \rho_j T_j V / B_j$. Для решения задачи (1)-(4): 1) Задаем размер первого размещаемого файла V и количество размещаемых файлов m ; 2) Формируем векторы $C(n)$, $X(n)$, $W(n)$: $C_j = \rho_j T_j V / B_j$; $X_j = 0$; $W_j = 0$; 3) Полагаем $i=1$; $T=0$; 4) Пока $i \leq m$: а) Формируем вектор $E(n)$: $E_j = 1$, если $W_j + V \leq B_j$, иначе $E_j = 0$; б) Среди узлов, для которых $E_j = 1$, находим узел p такой, что $C_p = \max_{E_j=1} C_j$; в) Помещаем файл в узел p : $X_p := 1$; $W_p := W_p + V$; г) Находим $T = T + C_p$; д) Задаем размер нового файла V ; е) Формируем новые векторы $C(n)$, $X(n)$: $C_j = \rho_j T_j V / B_j$; $X_j = 0$; ж) Полагаем $i=i+1$ и переходим к пункту а). На этапе 4а может оказаться, что файл не помещается ни в один из узлов, т.е. алгоритм не может решить задачу при заданных начальных условиях. В этом случае необходимо увеличить объем узлов, а затем повторить вычисления с шага 1. Временная сложность алгоритма - $O(m(2n-1))$.

Обозначим: P – наилучшее значение целевой функции, полученное полным перебором; A – значение целевой функции, полученное предлагаемым в статье алгоритмом; Q – относительная погрешность алгоритма: $Q = (P - A) \cdot 100\% / P$. Так как $P \geq A > 0$, то $0 \leq Q < 100\%$. Обозначим: A – значение целевой функции, полученное предлагаемым алгоритмом; M – максимально возможное значение целевой функции; Q_m – максимально возможная относительная погрешность алгоритма: $Q_m = (M - A) \cdot 100\% / M$. Значение M будет достигнуто, если файл размещается в узел p , такой, что $C_p = \max_j C_j, j=1...n$. Если задача имеет большую размерность, полным перебором значение P найти невозможно. Уточним оценку Q для такого случая. Будем считать, что P – значение равномерно распределенной случайной величины ξ , которая принимает значения в интервале $[A; M]$. Наиболее вероятное значение (мода) этой случайной величины совпадает с ее математическим ожиданием. Найдем математическое ожидание m_ξ , дисперсию

d_ξ , среднеквадратичное отклонение σ_ξ , средневероятное отклонение Δ_ξ случайной величины ξ по формулам:
 $m_\xi = (A + M)/2$; $d_\xi = (M - A)^2 / 12$; $\sigma_\xi = \sqrt{d_\xi}$; $\Delta_\xi = (M - A)/4$. Обозначим:
 $R = m_\xi + \Delta_\xi$; $Q_r = (R - A) \cdot 100\% / R$. Таким образом, наиболее вероятно, что
 $A \leq P \leq R \leq M$, следовательно $0 \leq Q \leq Q_r \leq Q_m < 100\%$.

Для исследования работы алгоритма проведен эксперимент на ПЭВМ AMD 500MHz. Программа составлена на Delphi 3.01, операционная система Windows 98. Были решены 5 задач размещения файлов узлам сети. Элементы матрицы F_{ij} в задачах формировались функцией $\text{gandom}(100)$; $\theta_j = 0,01$; $T_j = 100$. Вектор V сформирован по формулам: $V_1 = m$; $V_i = V_1 - (i - 1)$, где $i = 1 \dots m$. $B_1 = \sum_{i=1}^m V_i / n$; $B_j = B_1 + (j - 1)$, $j = 1 \dots n$.

Обозначим: T_p - время решения задачи алгоритмом. Результаты экспериментов показаны в таблице 1. С увеличением размерности задачи время работы алгоритма растет полиномиально. Относительная погрешность алгоритма не превышает 1%.

Таблица 1 – Результаты экспериментов

№	m	n	M	A	Q_r , %	Q_m , %	T_p , с.
1	10^4	10	903,77	897,32		0,7	0,28
2	$5 \cdot 10^4$	10	903,44	898,87		0,51	1,16
3	10^5	10	903,4	899,52		0,43	2,25
4	$5 \cdot 10^5$	10	903,8	900,36		0,38	11,21
5	10^6	10	903,88	900,52		0,37	22,47

Выводы. В статье сформулирована задача размещения файлов распределенной САПР по узлам вычислительной сети. В качестве оптимизируемого критерия размещения файлов предложена максимизация суммарного времени использования файлов узлов. Ограничениями в задаче служат объемы памяти и коэффициенты загрузки узлов. В статье предложен эвристический алгоритм решения задачи. Временная сложность алгоритма - $O(m(2n - 1))$, относительная погрешность Q - $0 \leq Q \leq Q_r \leq Q_m < 100\%$. Результаты экспериментов подтверждают эффективность алгоритма при решении задач большой размерности.

Список литературы

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. 44-49.
2. Рахлин В.А. Моделирование машин баз данных распределенной архитектуры. //Программирование, 1996, N 2. - С. 7-16.