

СЛУЧАЙНЫЙ ПОИСК ОПТИМАЛЬНОГО РАЗМЕЩЕНИЯ ФАЙЛОВ СРЕДИ УЗЛОВ КОМПЬЮТЕРНОЙ СЕТИ

Бельков Д.В.

Донецкий национальный технический университет, г. Донецк
кафедра вычислительной математики и программирования

Abstract

Belkov D.V. The random search for optimal file placement in the computer network. An important practical task, which consists in the optimal file placement in the computer network, is decided in this article. The random search for this task is proposed.

Общая постановка проблемы

При проектировании распределенных систем необходимо стремиться к рациональному размещению файлов по узлам компьютерной сети. Если задача рационального размещения файлов не будет решена при проектировании, то это может привести к перегрузке каналов связи и неэффективной работе распределенной системы. Поэтому одной из целей проектирования является синтез таблицы размещения файлов. Оптимизация размещения файлов имеет важное практическое значение как способ повышения эффективности функционирования компьютерных сетей за счет ограничения сетевого трафика по перегруженным каналам. Размещение файлов в распределенных системах не автоматизировано. Задача решается администратором системы. Интуитивное, основанное только на опыте администратора, решение задачи может значительно отличаться от оптимального.

Задача распределения файлов по узлам компьютерной сети относится к классу NP-трудных. В случае m файлов и n узлов, число вариантов распределения файлов по узлам составляет n^m . Для больших сетей анализ вариантов точными методами не возможен ввиду значительной трудности вычислений. Точные методы можно применять только для решения задачи малой размерности. В случае большой размерности необходимо использовать приближенные методы. Большинство работ, посвященных оптимальному размещению файлов, основано на применении направленного перебора. Предложенные в них методы не позволяют получать рациональные решения при большом количестве файлов, отличаются высокой трудоемкостью. Поэтому необходима разработка новых приближенных методов размещения файлов [1,2,3].

Постановка задач исследования

Для рационального размещения файлов распределенной системы среди узлов компьютерной сети необходимо решить следующие основные задачи:

- сформулировать задачу размещения файлов;
- разработать алгоритм решения задачи размещения файлов;
- исследовать эффективность работы алгоритма размещения файлов.

Решение задач и результаты исследований

Исходные данные задачи размещения файлов показаны в таблице 1.

Пусть $X_{ij} = 1$, если i -й файл расположен в j -м узле, иначе — $X_{ij} = 0$. Введем условные обозначения, показанные в таблице 2.

Коэффициент заполнения α_{ij} характеризует объем памяти j -го узла, оставшийся свободным после размещения i -го файла в этот узел. Переполнение узлов недопустимо, поэтому $\alpha_{ij} \leq 1$. Для экономии памяти коэффициент заполнения узлов необходимо максимизировать.

Таблица 1 — Исходные данные задачи размещения файлов

1	Объемы памяти узлов	$B_j, j=1,2,\dots,n$
2	Объемы файлов	$V_i, i=1,2,\dots,m$
3	Число копий файлов	$R_i, i=1,2,\dots,m$
4	Интенсивность запросов к файлам из узлов	$F_{ij}, i=1,2,\dots,m, j=1,2,\dots,n$
5	Число промежуточных узлов на наименее нагруженном маршруте из узлов к файлам	$K_{ij}, i=1,2,\dots,m, j=1,2,\dots,n$
6	Время обработки файлов в узлах	$T_{ij}, i=1,2,\dots,m, j=1,2,\dots,n$

Таблица 2 — Условные обозначения

1	Коэффициент заполнения узлов	$\alpha_{ij} = V_i / B_j, i=1,2,\dots,m, j=1,2,\dots,n$
2	Коэффициент использования узлов	$\rho_{ij} = F_{ij} T_{ij}, i=1,2,\dots,m, j=1,2,\dots,n$
3	Маршрутный коэффициент	$\beta_{ij} = 1/(1 + K_{ij}), i=1,2,\dots,m, j=1,2,\dots,n$
4	Уровень (количество) запросов к файлам из узлов	$L_{ij} = \alpha_{ij} \beta_{ij} \rho_{ij}, i=1,2,\dots,m, j=1,2,\dots,n$

Коэффициент использования ρ_{ij} является индикатором перегрузки узлов при обработке запросов к файлам. Очереди запросов возникают, если $\rho_{ij} \geq 1$. Для предотвращения перегрузки можно регулировать время обработки файлов в узлах T_{ij} так, чтобы выполнялось условие $0 < \rho_{ij} < 1$.

Маршрутный коэффициент β_{ij} характеризует маршрутизацию запросов к файлам. Число промежуточных узлов на любом маршруте не превышает значения $(n-1)$ поэтому $1/n \leq \beta_{ij} \leq 1$. Для ускорения обработки запросов за счет сокращения длины их маршрутов коэффициент β_{ij} необходимо максимизировать. Наименее нагруженные маршруты между узлами могут быть найдены сетевыми процессорами.

Уровень запросов к файлам характеризует объем сетевого трафика. Для повышения эффективности функционирования сети за счет снижения времени отклика можно максимизировать уровень локальных запросов в сети, решив задачу рационального размещения файлов по узлам (1)–(3).

Максимизация уровня локальных запросов в сети:

$$\sum_{i=1}^m \sum_{j=1}^n L_{ij} X_{ij} \rightarrow \max \tag{1}$$

Ограничения на число копий файлов:

$$1 \leq \sum_{j=1}^n X_{ij} \leq R_{ij}, X_{ij} \in \{0,1\} \tag{2}$$

Ограничения на объемы памяти узлов:

$$\sum_{i=1}^m V_i X_{ij} \leq B_j \tag{3}$$

Задача размещения файлов по узлам компьютерной сети является задачей линейного программирования с булевыми переменными. В работе [4] для решения задачи (1)–(3) предложен жадный алгоритм, который состоит из двух этапов. На первом этапе находятся для файла i те узлы, в которые файл помещается по объему. На втором этапе среди найденных узлов определяется узел с наибольшим значением L_{ij} и файл размещается в этот узел. Временная

сложность алгоритма — $O(n \sum_{i=1}^m R_i)$.

Для повышения точности решения задачи (1)–(3) в данной работе предлагается алгоритм случайного поиска. На каждой итерации матрице L ставится в соответствие матрица случайных величин γ , равномерно распределенных на интервале $[0;1]$. Очередной файл распределяется в тот узел, куда он помещается по объему и где значение γ максимально. Решением задачи является вариант размещения файлов, полученный на итерации с максимальной целевой функцией.

Для исследования работы предлагаемого алгоритма проведены серии вычислительных экспериментов. В каждом эксперименте алгоритмом случайного поиска решалась задача распределения m файлов без копий среди n узлов сети, $m=8, n=3$.

Известно [5,6], что интенсивность запросов к файлам в Internet меняется в пределах от 0 до 1000 запросов в секунду и подчиняется вероятностному распределению Парето

$$F(x) = P(X \leq x) = \begin{cases} 0, & x < k \\ 1 - (k/x)^\alpha, & x \geq k, \alpha < 2 \end{cases}, \text{ где } P(X \leq x) \text{ — вероятность того, что значение}$$

случайной величины X не превысит заданное число x . Параметр $\alpha = \alpha_1 = 1,5$. Большинство файлов находится в диапазоне 100 – 100 000 байт. Распределение объемов файлов подчиняется вероятностному распределению Парето с параметром $\alpha_2 = 1,1$.

Вычислительный эксперимент по размещению файлов среди узлов компьютерной сети проведен для случая $E_1 = 500$ запросов в секунду, $k_1 = (\alpha_1 - 1) \cdot E_1 / \alpha_1 = 166,67$ запросов в секунду. Интенсивность запросов к i -му файлу из j -го узла вычислялась по формуле: $F_{ij} = k_1 / (1 - \gamma)^{1/\alpha_1}$ (запросов в секунду), где γ — случайная величина, равномерно распределенная на интервале $(0;1)$, $E_2 = 50000$ байт, $k_2 = (\alpha_2 - 1) \cdot E_2 / \alpha_2 = 4545,46$ (байт). Объем i -го файла вычислялся по формуле: $V_i = k_2 / (1 - \gamma)^{1/\alpha_2}$ (байт), $i=1,2,\dots,8; j=1,2,3$.

Для сравнения результатов использовались решения, полученные в среде Excel с помощью мастера поиска решений [7], а также жадным алгоритмом [4] и полным перебором, которые были реализованы в среде Delphi. Эксперименты проводились на компьютере AMD 900 MHz 128 MB RAM.

В серии экспериментов решено 10 задач размещения m файлов по n узлам при $m=8, n=3$. Результаты решения жадным алгоритмом показаны в таблице 3. Средняя относительная погрешность решения, полученного жадным алгоритмом, составляет 9 %.

Результаты решения в среде Excel показаны на рисунке 1. Получено строго точное решение задачи, равное 1211,767. Динамика (в среднем по задачам) поиска решений случайным алгоритмом показана на рисунке 2. Строго точное решение определялось случайным поиском в среднем на 13 итерации. Методом полного перебора решение найдено на 6561 итерации. В среднем алгоритм случайного поиска работает лишь на 20 миллисекунд медленнее, чем жадный алгоритм.

Таким образом, алгоритм случайного поиска является наиболее рациональным по точности и быстродействию. Жадный алгоритм лучше случайного по быстродействию, но хуже по точности. Метод полного перебора вариантов решений имеет очень большое время работы и может применяться лишь при решении задач малого размера.

Таблица 3 — Результаты решения задачи жадным алгоритмом

№	Результат, полученный жадным алгоритмом	Относительная погрешность, %
1	2	3
1	1200,768	9,08
2	1199,896	9,8
3	1210,004	1,45
4	1205,346	5,3
5	1197,893	11,45
6	1209,114	2,19
7	1207,175	3,79
8	1099,653	10
9	1176,996	28,69
10	1201,242	8,69

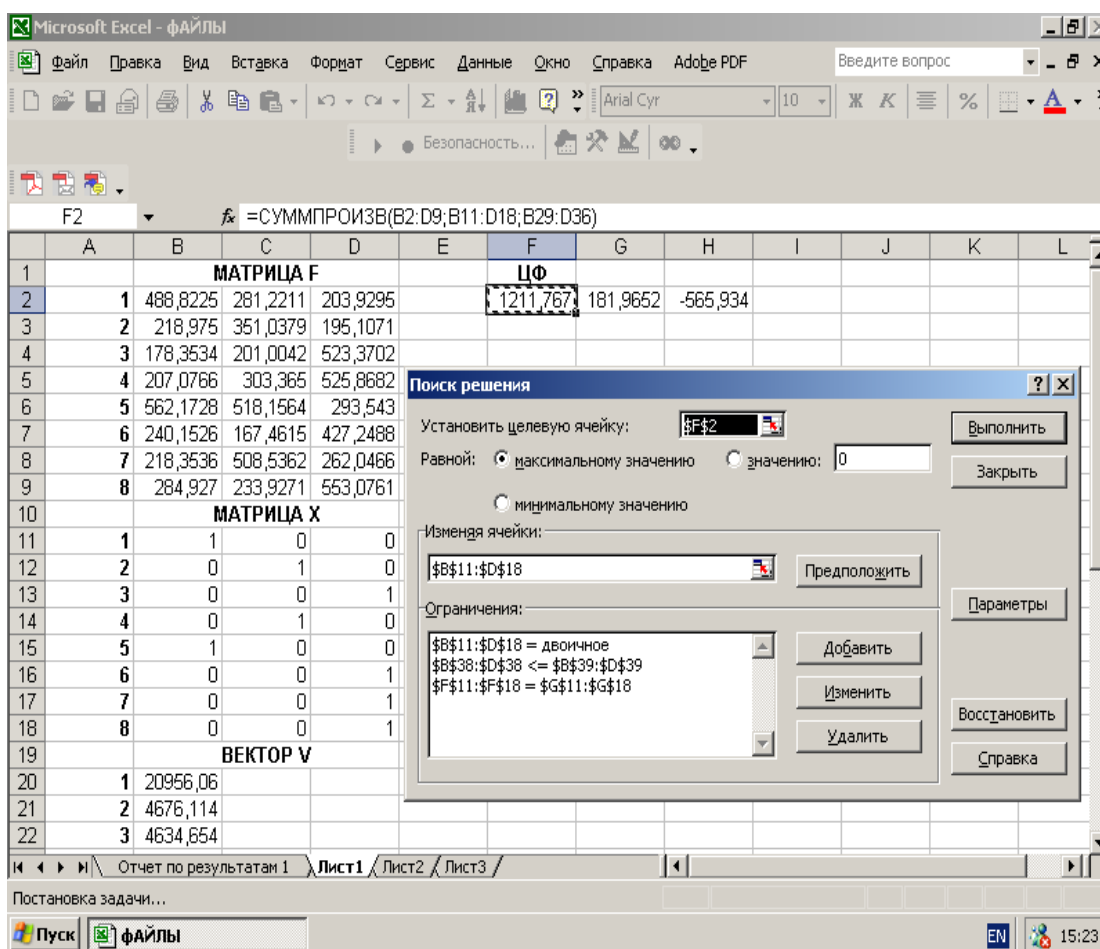


Рисунок 1 — Решение задачи размещения файлов средствами Excel

ДИНАМИКА ПОИСКА РЕШЕНИЙ СЛУЧАЙНЫМ АЛГОРИТМОМ

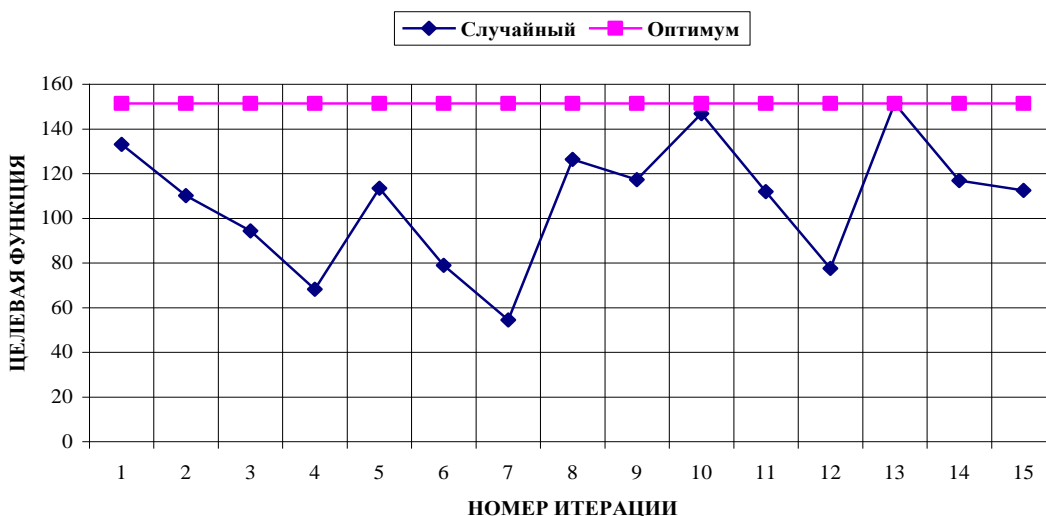


Рисунок 2 — Динамика поиска решений случайным алгоритмом

Выводы

1. Сформулирована задача размещения файлов в компьютерных сетях с учетом объемов памяти узлов и маршрутизации запросов к файлам как задача линейного программирования с булевыми переменными.

2. Разработан алгоритм случайного поиска оптимального размещения файлов в компьютерных сетях, позволяющий решить задачу более точно, чем жадный алгоритм.

3. Вычислительный эксперимент показал, что строго точное решение определялось случайным поиском в среднем на 13 итерации. Методом полного перебора решение найдено на 6561 итерации.

Перспективным направлением исследований является тестирование предложенного алгоритма при решении задач большого размера.

Литература

1. Mei A., Mancini L.V., Jajodia S. Secure dynamic fragment and replica allocation in large-scale distributed file systems. //IEEE Transactions on parallel and distributed systems. — 2003. — № 2. — P. 885–896.
2. Демидович О.В. Математичні моделі оптимального розподілу інформаційних ресурсів серед вузлів обчислювальних мереж та методи їх реалізації. Автореферат дисертації. Львів: 2001. — 20 с.
3. Тичковский Р.О., Цегелик Г.Г. Математичні моделі оптимального розподілу файлів серед вузлів обчислювальних мереж та методи їх реалізації. Третья Всеукраїнська конференція молодих науковців “Інформаційні технології в науці, освіті і техніці” (ІТОНТ – 2002). Черкаси: ЧДУ. — 2002. — С. 262–265.
4. Бельков Д.В. Методи і обчислювальні структури для розміщення файлів в комп’ютерних мережах. Автореферат дисертації. Донецьк: 2004.—21 с.
5. Менаске Д., Алмейда В. Производительность Web-служб. М.: DiaSoft. — 2003. — 465 с.
6. Труб И. Алгоритмическое обеспечение распределенных WEB-серверов. Открытые системы. — 2003. — № 5. — С.49–54.
7. Леоненков А.В. Решение задач оптимизации в среде MS EXCEL. СПб: БХВ-Петербург, 2005. — 690 с.