ОПТИМИЗАЦИЯ СОСТАВА И СТРУКТУРЫ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ МЕТОДА СРЕДНИХ

Фельдман Л. П., Михайлова Т.В.
Кафедра ПМиИ ДонНТУ
feldman@ r5.dgtu.donetsk.ua

Abstract

Feldman L., Michailova T. Structures optimization for high- performance systems of method middle.. *Approach for decision of optimization task for computer systems, that use of method middle, is presented. This method is more economical on calculations complication.*

Введение

В работе авторов [1] аналитически решена задача оптимизации состава и структуры вычислительной системы (ВС), представленной замкнутой стохастической сетью, обрабатывающих постоянное число задач. В работе [2], являющейся продолжением статьи [1], рассматривается общая постановка задачи оптимизации ВС, представленной замкнутой стохастической сетью, обрабатывающих постоянное число задач, и рассмотрен более эффективный подход для решения таких задач с использованием градиентных методов. В данной работе предлагается еще один подход для решения задачи оптимизации ВС с использованием метода средних, который по сравнению с предыдущими методами более экономичный по сложности вычислений.

1. Марковская модель системы клиент-сервер

Структура системы клиент- сервер приведена на рис. 1. В ней M рабочих станций пользователей и N групп серверов (количество которых $k_1,\ldots,k_N$, соответственно). В зависимости от вида ресурса, которым владеет сервер (файловая система, база данных, принтеры, процессоры или прикладные пакеты программ), определяется тип сервера, например файл-сервер, сервер базы данных, принт-сервер, вычислительный сервер, сервер приложений и т.д.[3].

Предполагается, что все рабочие станции заняты пользователями, каждый из которых может послать только один запрос на один из серверов. Пользователь, пославший запрос к одному из серверов, ждет ответа и, только после получения ответа может послать новый запрос.

В системе постоянно находится M запросов, часть из них m_1 находится у пользователей, а m_i - на i-ой группе серверов ($m_1+m_2+..+m_N=M$).

Функционирование рассматриваемой системы можно представить замкнутой стохастической сетью, содержащей N систем массового обслуживания (СМО), в которой циркулирует M заявок.

Граф передач этой сети изображен на рис. 2, на котором введены следующие обозначения:

$S_1, S_2,\ldots,S_k,\ldots, S_N$ – СМО, соответствующие клиентам и серверам;

S_0- фиктивная система, введенная для фиксации событий завершения задач пользователями;

p_{ij} - вероятности поступлений заявок из i-й СМО в j-ю,
p_{10}- вероятность завершения задачи пользователем.

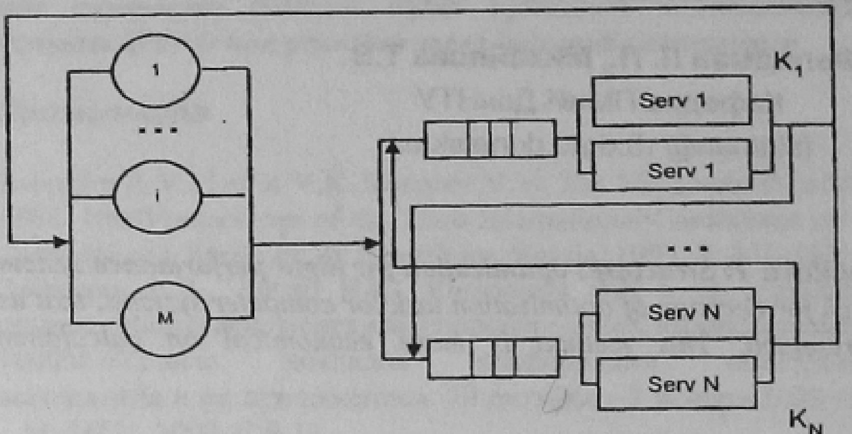

Рисунок 1 - Структура системы клиент – сервер

Поскольку в ВС находится постоянное число задач, то предполагается, что после завершения очередной задачи пользователь приступает к следующей. На рис. 2 этому соответствует передача через систему S_0. По графу передач определяются соотношения интенсивностей потоков заявок λ_i, поступающих в каждую из систем, и среднее число этапов обслуживания задачи пользователем и сервером $\alpha_i = \lambda_i / \lambda_0$ (i=1,…,N) [2].

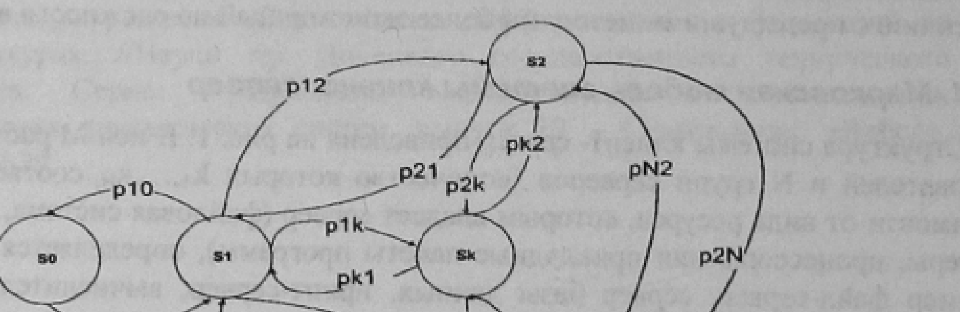

Рисунок 2 - Граф передач сети клиент- сервер

2. Задача оптимизации состава и структуры высокопроизводительных вычислительных систем

Рассматривается решение следующей задачи: определить быстродействие рабочих станций V_1 и серверов V_2 ,…,V_N, обеспечивающих минимальное время решения задачи U так , чтобы стоимость системы с М рабочими станциями не превышала заданного значения S*. Таким образом, необходимо найти минимум функции U, определяемой формулой (5). При этом надо удовлетворить ограничениям

$$c_1 M V_1 + c_2 V_2 k_2 + \ldots + c_N V_N k_N \leq S^*, \qquad V_i > 0,\ i=1,\ldots,N, \tag{1}$$

где c_i- стоимость единицы производительности устройства типа i,

k_i- количество серверов i-го вида.

Для решения рассматриваемой задачи можно применить метод множителей Лагранжа, т.е. найти минимум функции

$$G=U+\gamma(c_1MV_1+c_2V_2k_2+\ldots+c_NV_Nk_N -S^*), \tag{2}$$

где γ- неопределенный постоянный множитель. В этом случае V_k (k=1,…,N) и γ определяются как решение системы нелинейных уравнений

$$\frac{\partial G}{\partial V_1}=0,\ldots,\frac{\partial G}{\partial V_k}=0,\ldots,\frac{\partial G}{\partial V_N}=0, \quad c_1MV_1+c_2V_2k_2+\ldots+c_NV_Nk_N \le S^*. \tag{3}$$

Общее время решения задачи U вычисляется как сумма произведений средних времен пребывания в i-ой СМО ($u_i=\frac{m_i^* v_i}{\rho_i}$) на коэффициент посещений. Основные характеристики сети выражаются через стационарные вероятности [4], некоторые из которых имеют вид:

загрузка сервера $\rho_2=1-\sum_{i=1}^{M}\pi(i,0,M-i)$; среднее число задач, находящихся на сервере:

$$m_3^*=1-\sum_{j=1}^{M-i} j\pi(i,M-j-i,j),\ i=\overline{1,M}\ .$$

Стационарная вероятность вычисляется по следующей формуле [4]

$$\pi(m_1,m_2,\ldots,m_N)=\frac{\prod_{j=1}^{N}R_j(m_j)(\alpha_j v_j)^{m_j}}{\sum\limits_{\text{по всем состояниям}}\prod_{j=1}^{N}R_j(m_j)(\alpha_j v_j)^{m_j}},$$

$$\text{где}\quad R_j(m_j)=\begin{cases}\{1/m_j, & \text{если}\quad m_j\le k_j\\ \{1/(k_j!\,k_j^{\,m_j-k_j}), & \text{если}\quad m_j>k_j\end{cases}.$$

Выражение для функции U достаточно громоздкое, поэтому для выполнения преобразований, необходимых для получения системы (3), используется пакет *Mathematica*. Программа формирования и решения уравнений (3) подробно описывается в [1].

Полученная таким образом система нелинейных уравнений (3) на современных ПЭВМ решается при ограниченном значении M (M<10), т.к. функция U (5) использует стационарные вероятности (количество которых увеличивается с ростом количества клиентов M). В методе средних [5,6] используются рекуррентные формулы для определения времени решения задачи (5), что позволяет упростить решение поставленной задачи.

3. Метод средних

Основа этого метода - в последовательном вычислении характеристик (4)-(6), начиная с M=0 и до заданного значения [6].

Среднее время пребывания в i-й СМО:

$$u_i(M)=\nu_i\left[1+n_i(M-1)\right]+\sum_{j=1}^{M} j\left[\frac{1}{\mu_i(j)}-\nu_i\right]P_i(M-1,j-1), \tag{4}$$

где $\mu_i(m_i)=\begin{cases} m_i/\nu_i, & \text{если } m_i \le k_i, \\ k_i/\nu_i, & \text{если } m_i > k_i, \end{cases}$

$$P_i(M,l)=\frac{\lambda_i(M)}{\mu_i(l)}P_i(M-1,l-1),$$

$$P_i(0,0)=1.$$

Время решения задачи:

$$U(M)=\sum_{i=1}^{N}\alpha_i u_i(M). \tag{5}$$

Среднее число задач, находящихся в i- й СМО:

$$n_i(M)=\frac{M\alpha_i}{U(M)}u_i(M), \tag{6}$$

$$n_i(0)=0.$$

Интенсивность входного потока в i-ю СМО:

$$\lambda_i(M)=\lambda_0(M)*\alpha_i. \tag{7}$$

4. Метод оптимизации состава и структуры высокопроизводительных вычислительных систем

Предлагается следующий алгоритм оптимизации состава и структуры высокопроизводительных вычислительных систем.

1.Выбирается исходный вектор производительности устройств V_k (k=1,…,N) из условия (1). Задаются начальные значения величины шага h, времени отклика Uopt=∞, признак определения лучшей точки flag=0 (flag=1, если определена лучшая точка).

2.Методом средних считаются основные характеристики: средние времена пребывания в i-ой СМО, время отклика задачи U(M), среднее число задач, находящихся в i-ой СМО.

3.Если U(M) < Uopt, запоминаем новый вектор V_k (k=1,…,N), новый рекорд по времени отклика U(M) = Uopt, flag=1 (определена лучшая точка на данном шаге).

4.Если нет лучшей точки с шагом h (flag=0), уменьшаем его h=h/2.

5.Если h>ε, переходим к п.5, иначе конец алгоритма.

5.Строим новый вектор V_k (k=1,…,N)= V_k (k=1,…,N)+h из условия (1). Переходим к п.2.

Так как функция U в заданной области монотонна (что показано в [2]), минимум её достигается на границе области допустимых решений [6]. Градиентным методом в силу специфичности функции (2) минимум не всегда достигается. Поэтому методом координатного спуска [5] на границе уточняется результат [2]. Начальным приближением для этого метода используется точка, полученная градиентным методом. Затем на границе с единичным шагом выбирается направление координаты, по которой функция улучшается, и вычисляется новая точка. Если нет лучшей точки, шаг уменьшается. Процесс продолжается, пока не достигнется заданная точность.

5. Пример проектирования вычислительной системы

Спроектировать вычислительную систему клиент-сервер, обеспечивающую обработку потока поступающих запросов так, чтобы стоимость ВС была не больше $S^*=$ 120 тыс. грв, а время ответа было минимальным. Вычислительная система состоит из двух серверов и некоторого количества рабочих станций, быстродействие которых необходимо определить. Исходные данные: трудоемкость одного этапа решения задачи в миллионах операций $\theta_1=2$, $\theta_2=1$, $\theta_3=2$ количество этапов обслуживания $\alpha_1=20$, $\alpha_2=17.8$; $\alpha_3=16{,}6$; коэффициенты стоимости c1= 2 тыс.грв/ Мфлоп, c2= 5 тыс.грв /Мфлоп, c3=2 тыс.грв/Мфлоп. Результаты, полученные с использованием метода средних совпадают с результатами, полученными методами в [1,2] для M<10 (рис. 1,2), а для значений M>10 – производительности и время отклика вычислены только методом с использованием метода средних.

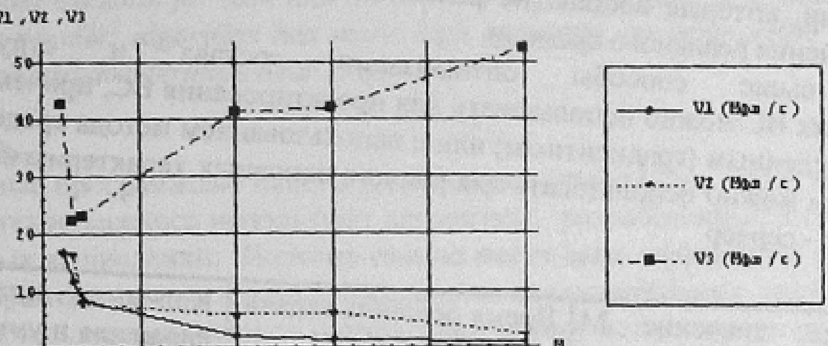

Рисунок 1 - Производительности рабочих станций и серверов

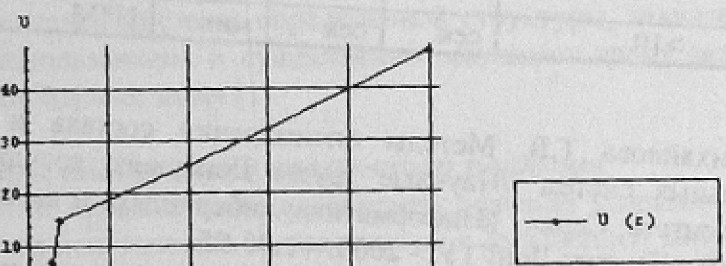

Рисунок 2 - Время отклика системы

6. Оценка трудоемкости метода

Чтобы вычислить значение времени отклика U (5) по теореме Джексона необходимо рассчитать стационарные вероятности и для каждой СМО среднее время пребывания в ней, загрузку и среднее число задач в этой СМО. Для вычисления нормировочной константы надо выполнить C*N*M*C_{M+N+1}^{N-1} операций сложения и умножения (где C-константа), в том числе M*C операций произведений сомножителей $R_j(m_j)(\alpha_j v_j)^{m_j}$ (так как $\sum_{j=1}^{N} m_j = M$), N операций сложения этих произведений; для вычисления стационарных вероятностей-C_{M+N+1}^{N-1} операций деления. Чтобы вычислить необходимые для расчета времени отклика основные характеристики требуется порядка M*N операций сложения. Сложность этого алгоритма – комбинаторная.

Одна итерация с использованием теоремы о среднем требует L*N*M операций сложения и умножения (где L - константа) и еще столько же операций для уточнения на границе. Сложность одной итерации этого алгоритма - полиномиальная.

Выводы

Рассмотрены два подхода к решению задачи синтеза: с использованием теоремы Джексона (аналитическое и численное решение системы (3) методами функций штрафа с уточнением методом покоординатного спуска) и с применением метода средних. Сравнительный анализ аналитического и численного решений системы (3) приведен в [2] в пользу численного решения.

Сравнение аналитического и с использованием теоремы о среднем методов приведен в таблице 1. В целом, алгоритм по теореме Джексона имеет комбинаторный порядок, а алгоритмы с использованием теоремы о среднем – полиномиальный, что позволяет решать задачи, которые вообще не решаются аналитическим методом на современных ЭВМ в течение реального времени.

Рассмотренные выше способы оптимизации состава и структур высокопроизводительных ВС можно использовать для проектирования ВС, причем при больших M – только численным (градиентным) или с использованием метода средних.

Также эти способы можно использовать для расчета основных характеристик (4)-(8) для системы клиент – сервер.

Таблица 1

Метод	Макс. значение M, при котором задача решается на ЭВМ	Время решения задачи при различных M			Кол-во операций сложения и умнож. (в целом)
		1	5	10	
т. Джексона	<10	сек	мин	часы	$N*M*C_{M+N+1}^{N-1}$
т. о среднем	>10	сек	сек	мин	N*M

Литература

1. Фельдман Л.П., Михайлова Т.В. Методы оптимизации состава и структуры высокопроизводительных систем //Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника»(ИКВТ-2000).- Донецк: ДонГТУ. - 2000. – с.90-95.
2. Фельдман Л.П., Михайлова Т.В. Способы оптимизации состава и структуры высокопроизводительных вычислительных систем //Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника»(ИКВТ-2001).- Донецк: ДонГТУ.- 2000. С. 80-85.
3. Шнитман В. Современные высокопроизводительные компьютеры. Информационно- аналитические материалы центра информационных технологий, 1996: http://hardware/app_kis.
4. Клейнрок Л. Вычислительные системы с очередями. – М.: Мир, 1979, 600с.
5. Фельдман Л.П., Дедищев В.А. Математическое обеспечение САПР. Моделирование вычислительных и управляющих систем. – Киев: УМК ВО, 1992, 256с.
6. Авен О. И. и др. Оценка качества и оптимизация вычислительных систем. – М.: Наука, 1982, 464с.

Поступила в редакційну колегію 1. 11.2001 р.