# ГЛАВА 3

# ЗАДАЧИ ОПТИМАЛЬНОЙ ОРГАНИЗАЦИИ КОНКУРИРУЮЩИХ ПРОЦЕССОВ ПРИ СОСРЕДОТОЧЕННОЙ ОБРАБОТКЕ

## 3.1. Эффективность и оптимальность организации конкурирующих процессов при достаточном числе процессоров

Одна из центральных проблем при разработке программного и аппаратного обеспечения многопроцессорных систем и вычислительных комплексов, а также при исследовании и создании математических моделей и методов анализа и организации параллельных процессов состоит в нахождении условий оптимальной реализации заданных объемов вычислений. Эта проблема в свою очередь порождает сложные комбинаторно–оптимизационные задачи по расчету как самих характеристик МС и ВК, так и задачи поиска критериев эффективности и оптимальности организации выполнения множества взаимодействующих процессов в условиях неограниченного и ограниченного параллелизма. Указанные задачи будем рассматривать в рамках математической модели сосредоточенной обработки конкурирующих процессов, введенной в главе 2, которая включает следующие параметры: $p$, $p \geq 2$ – число процессоров МС; $n$, $n \geq 2$ – число конкурирующих процессов; $s$, $s \geq 2$ – число блоков структурированного программного ресурса; $[t_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$ – матрица времен выполнения блоков программного ресурса конкурирующими процессами.

Кроме того, введем в рассмотрение параметр $\tau > 0$, характеризующий время дополнительных системных расходов, связанных с организацией параллельного использования блоков программного ресурса множеством конкурирующих процессов при сосредоточенной обработке.

Предполагается также, что все $n$ конкурирующих процессов используют только одну копию структурированного на блоки программного ресурса.

В дальнейшем нам понадобятся также следующие определения.

Определение 3.1. Систему конкурирующих процессов будем называть *однородной*, если времена выполнения *j–го* блока каждым из *i–х* процессов равны, т.е. $t_{ij} = t_j$, $i = \overline{1,n}$, $j = \overline{1,s}$.

Определение 3.2. Структурирование программного ресурса $\theta_s = (t_1,\ t_2,\ \ldots,\ t_s)$ для системы однородных конкурирующих процессов будем называть *равномерным*, если $t_1 = t_2 = \ldots = t_s = t$.

В главе 2 показано, что при достаточном числе процессоров, т.е. когда $n \le p$, минимальное общее время выполнения множества неоднородных конкурирующих процессов в точности определяется длиной критического пути в сетевом вершинно–взвешенном графе специального вида $G^c$, заданного матрицей $T^c = [t_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$. В общем случае, когда $n > p$, это время определяется длиной критического пути в сетевом графе специального вида, заданного матрицей $T_*^c$.

Кроме того, в разделе 2.5 показано, что для системы однородных процессов для вычисления минимального общего времени в асинхронном и первом синхронном режимах имеют место формулы (2.13), а во втором синхронном режиме – формулы (2.14), (2.15). Нетрудно показать, что аналогичные формулы вычисления минимального общего времени $T_{co}^{ac}(p,n,s,\tau)$, $T_{co}^{1}(p,n,s,\tau)$ и $T_{co}^{2}(p,n,s,\tau)$ имеют место и с учетом накладных расходов $\tau > 0$, с точностью до замены $T_c^s$ на $T_c^\tau$, $t_j$ на $t_j^\tau$, $t_{\max}^s$ на $t_{\max}^\tau$, где

$$T_c^\tau = \sum_{j=1}^{s} t_j^\tau, \quad t_j^\tau = t_j + \tau, \quad t_{\max}^\tau = \max_{1 \le j \le s} t_j^\tau .$$

В случае равномерного структурирования минимальное общее время $\overline{T}_c^\tau$ для всех трех режимов с учетом накладных расходов $\tau > 0$ также определяется по формулам (2.16) в точности до замены $t$ на $t^\tau = T_c^s / s + \tau$, $T_c^s = st$, т.е.

$$\overline{T}_c^\tau = \begin{cases} (n+s-1)t^\tau \ \textit{если} \ \ p \ge \min(n,s), \\ [(k+1)s + r - 1]t^\tau \ \textit{если} \ \ p < \min(n,s), \ \ n = kp + r, \ 1 \le r \le p. \end{cases}$$

Определение 3.3. Структурирование программного ресурса $\theta_s = (t_1,\ t_2,\ \ldots,\ t_s)$ для системы однородных конкурирующих процессов будем называть *эффективным* при фиксированных $p,\ n \ge 2$, если величина

$\Delta(s) = nT_c^s - T_{co}^{ac,1,2}(p,n,s,\tau) \geq 0$, где $nT_c^s$ – время выполнения $n$ процессов в последовательном режиме, $T_c^s = \sum_{j=1}^{s} t_j$.

Определение 3.4. Эффективное структурирование, при котором величина $\Delta(s)$ достигает наибольшего значения, будем называть *оптимальным*.

Покажем, что оптимальное структурирование достаточно искать среди класса равномерных структурирований программных ресурсов. Рассмотрим случай достаточного числа процессоров $(n \leq p)$.

В силу формул (2.13–2.14) и свойств функции взятия максимума вытекают необходимое условие эффективности и необходимое и достаточное условие оптимальности структурирования.

Теорема 3.1. *Для любого, не являющегося равномерным, эффективного структурирования программного ресурса* $\theta_s = (t_1,\ t_2,\ \ldots,\ t_s)$ *при* $n \leq p$ *существует равномерное более эффективное структурирование на s блоков.*

Теорема 3.2. *Для того, чтобы эффективное структурирование* $\theta_s = (t_1,\ t_2,\ \ldots,\ t_s)$ *программного ресурса для системы однородных конкурирующих процессов при* $n \le p$ *было оптимальным необходимо и достаточно, чтобы оно было равномерным.*

Доказательство. Согласно определения 3.3 условие эффективности для системы однородных конкурирующих процессов имеет вид:

$$\Delta_o(s) = (n-1)(T_c^s - t_{\max}^s) - (n+s-1)\tau \ge 0,$$

а в случае равномерного структурирования:

$$\overline{\Delta}(s) = (n-1)(T_c^s - \bar{t}) - (n+s-1)\tau \ge 0,$$

где $T_c^s = \sum_{j=1}^{s} t_j$, $\bar{t} = T_c^s / s$.

Покажем, что $\overline{\Delta}(s) > \Delta_o(s)$.

Рассмотрим разность $\overline{\Delta}(s) - \Delta_o(s) = (n-1)(t_{\max}^s - \bar{t})$. Случай, $\overline{\Delta}(s) - \Delta_o(s) \le 0$ будет иметь место только, если $\max\limits_{1 \le j \le s} t_j \le \bar{t}$. Последнее невозможно, так как из того, что структурирование не является равномерным, следовало бы $\sum_{j=1}^{s} t_j < s\bar{t} = T_c^s$, а по условию $\sum_{j=1}^{s} t_j = T_c^s$. Получили противоречие $T_c^s < T_c^s$, что и доказывает теорему 3.1.

Теорема 3.2 следует из теоремы 3.1 и определения 3.4.

Сформулируем далее необходимые и достаточные условия (критерий) существования эффективного структурирования программного ресурса на $s$, $s \geq 2$, блоков при достаточном числе процессоров в зависимости от величины накладных расходов $\tau$.

Теорема 3.3. *Для существования эффективного структурирования программного ресурса на s блоков для системы однородных конкурирующих процессов при* $n \leq p$ *и* $\tau > 0$, *необходимо и достаточно, чтобы выполнялись следующие условия:*

$$\tau \leq \begin{cases} f(1+\sqrt{n}), & \text{если } \sqrt{n} - \text{целое}, \\ \max\{f(1+[\sqrt{n}], f(2+[\sqrt{n}])\}, & \text{если } \sqrt{n} - \text{нецелое}, \end{cases}$$

*здесь* $f(x) = (n-1)T_c^s(x-1)/x(x+n-1)$, *а [x] – наименьшее целое, не превосходящее x.*

Доказательство. Необходимость следует из определения 3.3 и теоремы 3.1. Действительно, в силу определения 3.3 с учетом формул (2.16) условие эффективности равносильно следующему:

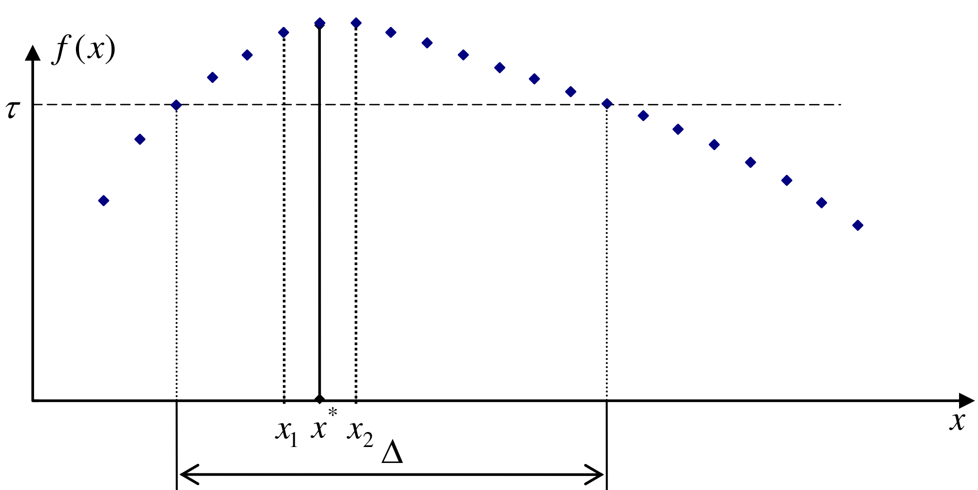
$$\tau \le \frac{(n-1)T_c^s(s-1)}{s(s+n-1)}. \tag{3.1}$$

Введя функцию $f(x) = (n-1)T_c^s(x-1)/x(x+n-1)$, $x > 0$, легко показать, что она достигает максимума при $x = 1+\sqrt{n}$. Выбрав в качестве эффективного структурирования на $s$ блоков, при котором $s = x = 1+\sqrt{n}$, если $\sqrt{n}$ – целое или $s = x \in \{1+[\sqrt{n}], 2+[\sqrt{n}]\}$, если $\sqrt{n}$ – нецелое, получим необходимость.

Достаточность следует из (3.1), так как $f(x)$ достигает наибольшего значения при $s = x = 1+\sqrt{n}$, если $\sqrt{n}$ – целое или при $s = x \in \{1+[\sqrt{n}], 2+[\sqrt{n}]\}$, если $\sqrt{n}$ – нецелое.

Замечание 3.1. *При* $p = s = 2$ *структурирование будет эффективным, если отношение* $\dfrac{\tau}{T_c^s} \le \dfrac{n-1}{2(n+1)}$.

На рис. 3.1 изображен график функции $y = f(x)$, $x > 0$,



при фиксированных $n$, $T_c^s$, $\tau$.

**Рис. 3.1.**

Существование эффективного структурирования определяется областью $\Delta$. Все целочисленные точки отрезка $\Delta$ являются значениями $s$, при которых структурирование будет эффективным, при этом $x_1 = 1 + [\sqrt{n}]$, $x^* = 1 + \sqrt{n}$, $x_2 = 2 + [\sqrt{n}]$.

Решение задачи об оптимальности равномерного структурирования программного ресурса на $s$ блоков для достаточного числа процессоров следует из теоремы 3.4.

Теорема 3.4. *Для того, чтобы эффективное структурирование программного ресурса $\theta_s$ на s блоков при $n \le p$ было оптимальным при заданных $T_c^s > 0$, $\tau > 0$, $n \ge 2$, необходимо и достаточно, чтобы оно было равномерным и число блоков s равнялось тому из чисел p,* $\left[\left[\sqrt{\frac{(n-1)T_c^s}{\tau}}\right], \left[\sqrt{\frac{(n-1)T_c^s}{\tau}}\right] + 1\right] \cap [2, p]$, *при котором функция* $\Delta(s)$ *принимает наибольшее значение.*

Доказательство. В силу теоремы 3.2 оптимальное структурирование следует искать среди равномерных. Тогда очевидно, что функция $\Delta(s)$ будет иметь вид:

$$\Delta(s) = (n-1)T_c^s(1-1/s) - (s+n-1)\tau .$$

Необходимость. Введем функцию действительного аргумента $x$ вида:

$$\Delta(x) = (n-1)T_c^s(1-1/x) - (x+n-1)\tau, \; x \geq 1 .$$

Согласно определению 3.4 структурирование будет оптимальным в той точке $x$, где $\Delta(x)$ достигает своего наибольшего значения. Покажем, что функция $\Delta(x)$ достигает своего наибольшего значения в точке $x = \sqrt{\frac{(n-1)T_c^s}{\tau}}$. Действительно,

$$\Delta'(x) = \frac{(n-1)T_c^s}{x^2} - \tau ,$$

$$\Delta''(x) = -\frac{2T_c^s(n-1)}{x^3} < 0 \text{, т.к. } x > 0,\; n \geq 2.$$

Следовательно, функция $\Delta(x)$ имеет максимум в точке, где $\Delta'(x) = 0$, т.е. $x^* = \sqrt{\frac{(n-1)T_c^s}{\tau}}$. Целочисленные точки, в которых достигается наибольшее значение функции $\Delta(s)$, будут $s = [x^*]$ или $s = [x^*]+1$. Следовательно, в качестве $s$

можно выбрать одно из чисел $\left[\sqrt{\frac{(n-1)T_c^s}{\tau}}\right]$, $\left[\sqrt{\frac{(n-1)T_c^s}{\tau}}\right]+1$, где функция $\Delta(s)$ принимает большее значение.

Если же окажется, что ни одна из точек $\left[\sqrt{\frac{(n-1)T_c^s}{\tau}}\right]$, $\left[\sqrt{\frac{(n-1)T_c^s}{\tau}}\right]+1$, в которой функция $\Delta(x)$ принимает наибольшее значение не принадлежит $[2, p]$, то в качестве оптимального структурирования по числу блоков выбираем $s^* = p$.

Достаточность следует из свойств выпуклости функции $\Delta(s)$ при $n \le p$ на $[2, p]$.

Следует заметить, что наряду со способом нахождения оптимального структурирования $s^*$ с помощью теоремы 3.4 известен также другой подход (см. 3.2), базирующийся на компоновке подряд идущих блоков. Однако, при использовании этого подхода, теорема 3.4 позволяет получить точную оценку отклонения от оптимума при вычислении минимального общего времени выполнения конкурирующих процессов.