

## ЗАДАЧА ОПТИМИЗАЦИИ ЧИСЛА ПРОЦЕССОРОВ ПРИ РАСПРЕДЕЛЁННОЙ ОБРАБОТКЕ

© 2011 П. А. Павлов

Полесский государственный университет

Решена задача определения временных характеристик распределённых процессов, конкурирующих за программный ресурс, и предложен метод нахождения оптимального числа процессоров, обеспечивающих директивное время выполнения заданных объемов вычислений.

*Распределённый процесс, программный ресурс, асинхронный режим, однородная распределённая система, неограниченный параллелизм, ограниченный параллелизм.*

**Введение.** Постоянное существование задач сверхвысокой сложности: задачи проектирования сложных систем; задачи оптимизационного плана развития экономики страны или отдельного региона, сооружений, технологических процессов; задачи эффективного использования спутников Земли для развития народного хозяйства и др., характеризуются большой размерностью, десятками сотен и миллионов независимых переменных и соответствующих ограничений. Указанные задачи можно эффективно решать, используя идеи распараллеливания сложных вычислительных процессов и обработки больших объёмов данных и знаний с помощью параллельных многопроцессорных систем (МС) и вычислительных комплексов (ВК), “объединяя в единое целое сведения из таких областей, как архитектура компьютеров и вычислительных систем, системное программирование и языки программирования, различные методы обработки информации” [1]. Принципы параллельной организации сложных процессов являются не только одним из универсальных способов достижения высокой производительности и надёжности вычислительных средств, они присущи системам управления, операционным системам, системам автоматизированного проектирования, промышленным процессам, конвейерным линиям и т. д. В этой связи возникает необходимость новых принципов организации вычислений и распределения ресурсов, создания эффективного аппа-

ратного и программного обеспечения, оптимального планирования и распределения вычислительных процессов [2]. При этом особую актуальность приобретают задачи эффективного управления множеством процессов, имеющих доступ к общим ресурсам, в том числе к программным. Математическая постановка такого рода задач была предложена и исследована ранее в работах [2–5].

В настоящей статье решена задача определения временных характеристик процессов вычислений, конкурирующих за программный ресурс при параллельной распределённой обработке в случае ограниченного параллелизма [2], и предложен метод нахождения минимального числа процессоров, обеспечивающих директивное время выполнения заданных объемов вычислений. Ввиду дискретного и комбинаторного характера математических задач параллельного программирования определённый прогресс на пути их решения был достигнут за счёт применения математического аппарата и методов дискретных систем и дискретной оптимизации, теории расписаний и сетевых графов, теории алгоритмов и множеств, алгебры матриц и др.

### 1. Основные понятия и определения.

Как и в [2–4], процесс будем рассматривать как последовательность блоков  $Q_1, Q_2, \dots, Q_s$ , для выполнения которых используется множество процессоров. При этом процесс будем считать *распределённым*, если все бло-

ки или часть из них выполняются на разных процессорах. Процессы, которые для ускорения выполнения обрабатываются параллельно, взаимодействуя путём обмена информацией, будем называть *кооперативными* или *взаимодействующими* процессами. Последовательность программных блоков, которую процессорам необходимо выполнять многократно, будем называть *программным ресурсом PR*, а множество соответствующих процессов – *конкурирующими*.

Математическая модель системы распределённой обработки конкурирующих процессов включает в себя [4]:  $s, s \geq 2$  – число блоков линейно-структурированного программного ресурса  $PR = (Q_1, Q_2, \dots, Q_s)$ ;  $n, n \geq 2$  – число распределённых относительно  $PR$  конкурирующих процессов;  $p, p \geq 2$  – число процессоров многопроцессорной системы; матрицу  $T = [t_{ij}]$  времен выполнения  $j$ -х блоков  $i$ -ми конкурирующими процессами  $i = \overline{1, n}, j = \overline{1, s}$ ;  $\varepsilon$  – время, характеризующее дополнительные системные расходы по организации структурирования и параллельного использования блоков  $PR$ .

*Определение 1.* Распределённая система  $n$  взаимодействующих конкурирующих процессов называется *неоднородной*, если времена выполнения блоков  $PR$  зависят от объёмов обрабатываемых данных и/или их структуры, т.е. разные для разных процессов.

Так же, как и в [3, 4], будем считать, что взаимодействие процессов, процессоров и блоков линейно-структурированного программного ресурса подчинено следующим условиям: 1) ни один из блоков  $PR$  не может обрабатываться одновременно более чем одним процессором; 2) ни один из процессоров не может обрабатывать одновременно более одного блока; 3) обработка каждого блока осуществляется без прерываний; 4) распределение блоков программного ресурса по процессорам  $MC$  для каждого из процессов осуществляется циклически по правилу: блок с номером  $j = kp + i, j = \overline{1, s}, i = \overline{1, p}, k \geq 0$  распределяется на процессор

с номером  $i$ ; 5) отсутствуют простои процессоров при условии готовности блоков, а также невыполнение блоков при наличии процессоров; 6) для каждого из  $n$  процессов момент завершения выполнения  $j$ -го блока на  $i$ -м процессоре совпадает с моментом начала выполнения следующего ( $j + 1$ )-го блока на  $(i + 1)$ -м процессоре,  $i = \overline{1, p - 1}, j = \overline{1, s - 1}$ ; 7) для каждого из блоков структурированного программного ресурса момент завершения его выполнения  $l$ -м процессом совпадает с моментом начала его выполнения  $(l + 1)$ -м процессом на том же процессоре,  $l = \overline{1, n - 1}$ .

*Асинхронный режим* взаимодействия процессоров, процессов и блоков предполагает отсутствие простоев процессоров  $MC$  при условии готовности блоков, а также невыполнение блоков при наличии процессоров и определяется условиями 1–5.

Условия 1–4, 6 определяют *первый синхронный режим*, обеспечивающий непрерывное выполнение блоков  $PR$  внутри каждого из процессов.

*Второй синхронный режим*, определяемый условиями 1–4, 7, обеспечивает непрерывное выполнение каждого блока всеми процессами.

**2. Неоднородные распределённые конкурирующие процессы в асинхронном режиме.** В [5] для вычисления минимального общего времени  $T_n^{ac}(p, n, s, \varepsilon)$  выполнения  $n \geq 2$  неоднородных распределённых конкурирующих процессов, использующих структурированный на  $s \geq 2$  блоков программный ресурс в многопроцессорной системе с  $p \geq 2$  процессорами с учётом параметра  $\varepsilon > 0$  в случае *неограниченного параллелизма* ( $2 \leq s \leq p$ ), был использован функционал задачи Беллмана–Джонсона, который имеет вид:

$$T_n^{ac}(p, n, s, \varepsilon) = \max_{1 \leq u_1 \leq u_2 \leq \dots \leq u_{s-1} \leq n} \left[ \sum_{i=1}^{u_1} t_{i1}^\varepsilon + \sum_{i=u_1}^{u_2} t_{i2}^\varepsilon + \dots + \sum_{i=u_{s-1}}^n t_{is}^\varepsilon \right],$$

где  $t_{ij}^\varepsilon = t_{ij} + \varepsilon$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ ,

а  $u_1, u_2, \dots, u_{s-1}$  – целые положительные числа.

Было также предложено графоаналитическое решение задачи определения  $T_n^{ac}(p, n, s, \varepsilon)$ . По заданным  $n, s$  и матрице времён выполнения блоков программного ресурса с учётом параметра  $\varepsilon$   $T^\varepsilon = [t_{ij}^\varepsilon]$ ,  $t_{ij}^\varepsilon = t_{ij} + \varepsilon$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$  построен сетевой вершинно-взвешенный граф  $G_1^{ac}$ , который содержит  $ns$  вершин, расположенных в узлах прямоугольной  $n \times s$ -решетки (рис. 1). Каждой вершине графа  $G_1^{ac}$  соответствует значение  $t_{ij}^\varepsilon$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ , причём  $t_{11}^\varepsilon$  – начальная вершина,  $t_{ns}^\varepsilon$  – конечная. Дуги в графе  $G_1^{ac}$  отражают линейный порядок выполнения блоков  $Q_j$ ,  $j = \overline{1, s}$  программного ресурса каждым из процессов, а также линейный порядок использования одних и тех же блоков разными процессами.

*Теорема 1. Минимальное общее время выполнения  $n$  ( $n \geq 2$ ) неоднородных распре-*

*делённых конкурирующих процессов, использующих структурированный на  $s$  ( $s \geq 2$ ) блоков программный ресурс с временами выполнения блоков, задаваемыми матрицей  $T^\varepsilon = [t_{ij}^\varepsilon]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ , в многопроцессорной системе с  $p$  ( $p \geq 2$ ), процессорами в асинхронном режиме в случае  $2 \leq s \leq p$  определяется длиной критического пути в сетевом вершинно-взвешенном графе  $G_1^{ac}$  из начальной вершины  $t_{11}^\varepsilon$  в конечную  $t_{ns}^\varepsilon$ .*

Рассмотрим случай *ограниченного параллелизма*, т.е. когда  $s > p$ ,  $s = kp + r$ ,  $k \geq 1$ ,  $1 \leq r < p$ . Для вычисления общего времени выполнения множества распределённых неоднородных конкурирующих процессов в асинхронном режиме всё множество блоков разобьём на  $k + 1$  группу по  $p$  блоков в каждой, за исключением  $k + 1$ -й группы, которая будет содержать  $r$  блоков. На рис. 2 представлен пример диаграммы Ганта, которая отражает выполнение  $n = 4$  неоднородных распределённых конкурирующих процессов, использующих структурированный на  $s = 8$  блоков  $PR$  в  $MC$  с  $p = 3$  процессорами. Времена выполнения блоков линейно-структу-

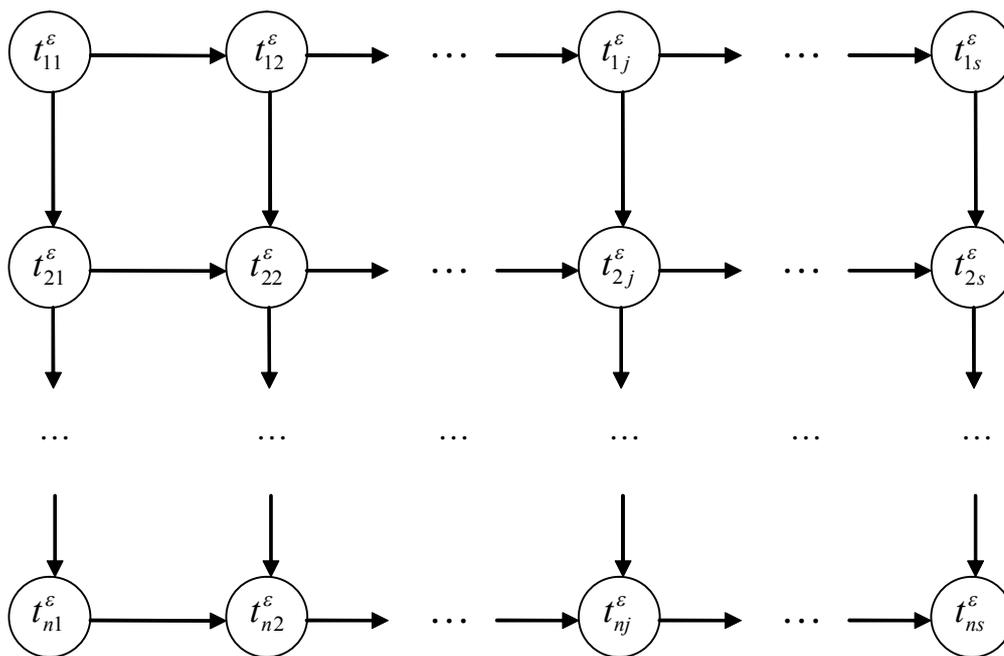


Рис. 1. Сетевой вершинно-взвешенный граф  $G_1^{ac}$

рированного программного ресурса проставлены над соответствующими блоками. Очевидно, что если выполнение очередной группы из  $p$  блоков начинать только после полного завершения выполнения предыдущей группы, то общее суммарное время выполнения всех  $n$  процессов в этом случае будет определяться как сумма длин критических путей в каждой из подряд идущих несовмещенных диаграмм Ганта, задаваемых прямой суммой матриц  $T_l^\varepsilon$ ,  $l = \overline{1, k+1}$ . Однако это время можно существенно сократить, если воспользоваться приёмом совмещения последовательных диаграмм Ганта по оси времени справа налево. Причём совмещение осуществляется поблочко, начиная со второй диаграммы, на максимально возможную величину таким образом, чтобы не нарушались технологические условия, определяющие асинхронный режим взаимодействия процес-

сов, процессоров и блоков. В результате совмещения получим результирующую совмещённую диаграмму Ганта (рис. 3).

Полученная структура результирующей совмещённой диаграммы Ганта будет полностью определяться матрицей  $T^*$  (1) и состоять из подматриц  $T_1^\varepsilon, T_2^\varepsilon, \dots, T_{k+1}^\varepsilon$  размерностью  $n \times p$  каждая. При этом подматрицы  $T_l^\varepsilon$ ,  $l = \overline{1, k+1}$  в результирующей матрице  $T^*$  располагаются таким образом, чтобы не нарушался характер взаимодействия как блоков  $PR$ , выполняемых одним и тем же процессом, так и блоков, выполняемых на одном и том же процессоре. Первая строка матрицы  $T^*$  будет состоять из подматриц  $T_l^\varepsilon$ ,  $l = \overline{1, k+1}$ , что отражает характер взаимодействия блоков программного ресурса внутри каждого из  $n$  процессов:

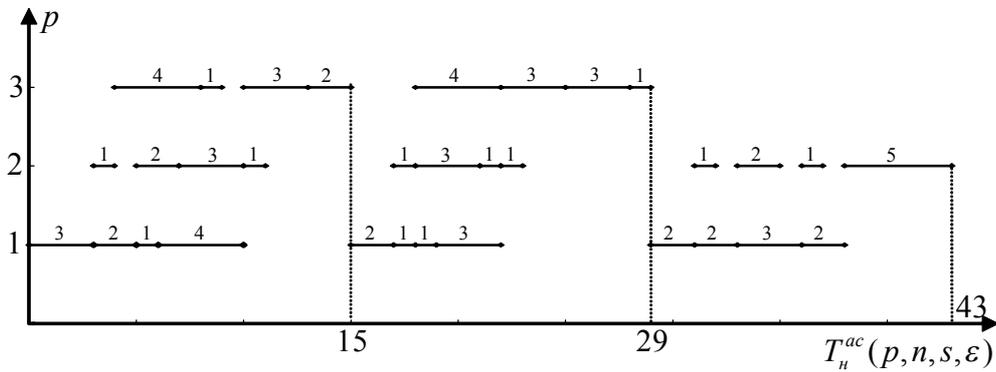


Рис. 2. Несовмещенная диаграмма Ганта

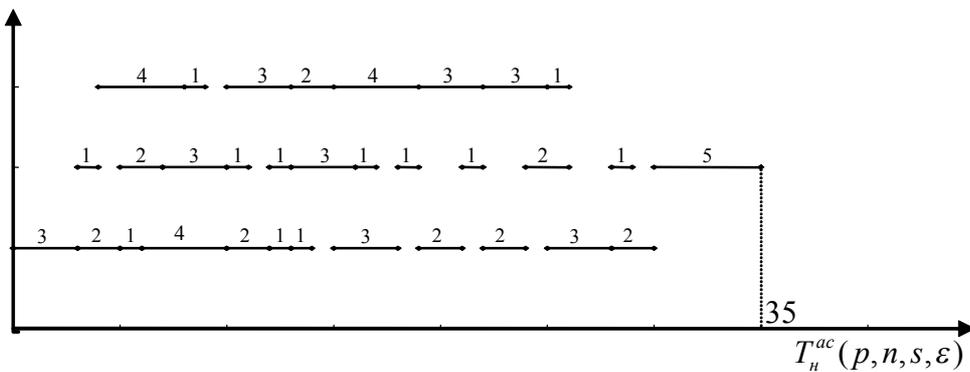


Рис. 3. Совмещенная диаграмма Ганта

$$T^* = \begin{bmatrix} T_1^\varepsilon & T_2^\varepsilon & T_3^\varepsilon & \dots & T_k^\varepsilon & T_{k+1}^\varepsilon \\ T_2^\varepsilon & T_3^\varepsilon & T_4^\varepsilon & \dots & T_{k+1}^\varepsilon & 0 \\ T_3^\varepsilon & T_4^\varepsilon & T_5^\varepsilon & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ T_k^\varepsilon & T_{k+1}^\varepsilon & 0 & \dots & 0 & 0 \\ T_{k+1}^\varepsilon & 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \quad (1)$$

где  $T_l^\varepsilon$  и  $T_{k+1}^\varepsilon$  матрицы вида

$$T_l^\varepsilon = \begin{bmatrix} t_{1,(l-1)p+1}^\varepsilon & t_{1,(l-1)p+2}^\varepsilon & \dots & t_{1,lp}^\varepsilon \\ t_{2,(l-1)p+1}^\varepsilon & t_{2,(l-1)p+2}^\varepsilon & \dots & t_{2,lp}^\varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ t_{n,(l-1)p+1}^\varepsilon & t_{n,(l-1)p+2}^\varepsilon & \dots & t_{n,lp}^\varepsilon \end{bmatrix}, \quad l = \overline{1, k},$$

$$T_{k+1}^\varepsilon = \begin{bmatrix} t_{1,kp+1}^\varepsilon & t_{1,kp+2}^\varepsilon & \dots & t_{1,kp+r}^\varepsilon & 0 & \dots & 0 \\ t_{2,kp+1}^\varepsilon & t_{2,kp+2}^\varepsilon & \dots & t_{2,kp+r}^\varepsilon & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_{n,kp+1}^\varepsilon & t_{n,kp+2}^\varepsilon & \dots & t_{n,kp+r}^\varepsilon & 0 & \dots & 0 \end{bmatrix}.$$

В матрице (1) очередная строка получается соответствующим смещением подматриц  $T_l^\varepsilon$ ,  $l = \overline{2, k+1}$  справа налево на максимальную величину, сохраняющую линейный порядок выполнения блоков программного ресурса на одном и том же процессоре.

С учётом того, что все подматрицы  $T_l^\varepsilon$ ,  $l = \overline{2, k+1}$  имеют одну и ту же размерность  $n \times p$ , величина смещения на каждом шаге будет равна  $p$ . Элементы смещённой на каждом шаге матрицы заполняются нулями. Выполнив таким образом  $k$  шагов смещений, получим структуру результирующей матрицы  $T^*$ . Отметим, что результирующая матрица  $T^*$  будет иметь размерность  $(k+1)n \times (k+1)p$ . Матрица  $T^*$  будет блочной, симметричной, верхне-диагональной относительно второй диагонали типа Ганкелевой порядка  $k+1$  [2]. По весам, задаваемым матрицей  $T^*$ , строится сетевой вершинно–

взвешенный граф  $G_2^{ac}$ . Вершины этого графа расположены в узлах прямоугольной  $(k+1)n \times (k+1)p$ –решётки. Как и в случае  $s \leq p$ , общее время  $T_n^{ac}(p, n, s = kp + r, \varepsilon)$  выполнения  $n$  неоднородных распределённых конкурирующих процессов определяется длиной критического пути из начальной вершины  $t_{11}^\varepsilon$  в конечную  $t_{(k+1)n, (k+1)p}^\varepsilon$ . Имеет место теорема.

*Теорема 2. Минимальное общее время  $T_n^{ac}(p, n, s, \varepsilon)$  выполнения  $n$  ( $n \geq 2$ ) неоднородных распределённых конкурирующих процессов, использующих линейно структурированный на  $s$  ( $s \geq 2$ ) блоков программный ресурс с временами выполнения блоков, задаваемыми матрицей  $T^\varepsilon = [t_{ij}^\varepsilon]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ , в многопроцессорной системе с  $p$  ( $p \geq 2$ ) процессорами и дополнительными системными расходами  $\varepsilon > 0$  в асинхронном режиме в случае  $s = kp + r$ ,  $k \geq 1$ ,  $1 \leq r < p$  определяется длиной критического пути из начальной вершины  $t_{11}^\varepsilon$  в конечную вершину  $t_{(k+1)n, (k+1)p}^\varepsilon$  сетевого вершинно–взвешенного графа  $G_2^{ac}$ .*

**3. Задача оптимизации числа процессоров при распределённой обработке.** Несомненно, время выполнения всех распределённых конкурирующих процессов  $T_n^{ac}(p, n, s, \varepsilon)$  будет существенно зависеть от количества имеющихся процессоров. Задача состоит в том, чтобы при заданных  $n, s, \varepsilon$ ,  $[t_{ij}^\varepsilon]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$  и заданного директивного времени выполнения всех распределённых конкурирующих процессов  $d$  найти минимальное число процессоров  $p^*$ , обеспечивающих директивное время выполнения. Решение данной задачи рассмотрим для общего случая *асинхронного режима*, т.е. когда процессы являются *неоднородными*.

Введём в рассмотрение двумерный массив  $M_q$  переменной длины, составленный

специальным образом из элементов матрицы  $[t_{ij}^\varepsilon]$ , где  $t_{ij}^\varepsilon = t_{ij} + \varepsilon$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ , а  $q, q \in N$  – порядковый номер результирующей матрицы времен выполнения блоков (двумерного массива переменной длины  $M_q$ ).

*Определение 2.* Число процессоров МС будем называть *достаточным* и обозначать  $p^s$  при заданных  $n, s$ , если  $p^s = s$ .

Обозначим через  $T_n^{ac}(p^s, n, s, \varepsilon)$  минимальное общее время выполнения множества конкурирующих процессов при достаточном числе процессоров  $p^s$ , а  $T_n^{ac}(p, n, s, \varepsilon)$  – минимальное общее время при исходном числе процессоров  $p$ . Имеет место теорема.

*Теорема 3.* При заданных  $n, s, \varepsilon, [t_{ij}]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$  в случае достаточного ( $p^s = s$ ) и ограниченного ( $p < s$ ) числа процессоров МС имеет место соотношение  $T_n^{ac}(p^s, n, s, \varepsilon) \leq T_n^{ac}(p, n, s, \varepsilon)$ .

Доказательство следует из того, что при достаточном числе процессоров ( $p^s = s$ ) минимальное общее время  $T_n^{ac}(p^s, n, s, \varepsilon)$  вычисляется как длина критического пути в сетевом вершинно–взвешенном графе  $G_1^{ac}$ . А при  $s > p$ ,  $s = kp + r$ ,  $k \geq 1$ ,  $1 \leq r < p$  минимальное общее время  $T(p < s, n, s, \varepsilon)$ , согласно теореме 2, определяется длиной критического пути в сетевом вершинно–взвешенном графе  $G_2^{ac}$ , задаваемом матрицей  $T^*$  специального вида (1) размерности  $(k+1)n \times (k+1)p$ .

Ввиду того, что исходная матрица  $T^\varepsilon = [t_{ij}^\varepsilon]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$  входит в качестве подматрицы в первую строку матрицы (1), сетевой вершинно–взвешенный граф  $G_1^{ac}$ , определяемый матрицей  $[t_{ij}^\varepsilon]$ ,  $i = \overline{1, n}$ ,

$j = \overline{1, s}$ , будет подграфом сетевого вершинно–взвешенного графа  $G_2^{ac}$ , определяемого матрицей (1). Так как длина критического пути в сетевом подграфе  $G_1^{ac}$  данного графа не может превышать длину критического пути в самом сетевом графе  $G_2^{ac}$ , то справедливо неравенство

$$T_n^{ac}(p^s, n, s, \varepsilon) \leq T_n^{ac}(p, n, s, \varepsilon),$$

что и требовалось доказать.

Следует также заметить, что в случае  $s < p$ , согласно принятой модели, достаточно использовать только  $s$  процессоров, а остальные  $p - s$  не будут задействованы.

Теорема 3 является отправной точкой для построения метода нахождения оптимального числа процессоров  $p^*$ , обеспечивающих директивное время  $d$  выполнения неоднородных конкурирующих процессов при распределённой обработке в условиях асинхронного режима их взаимодействия.

*Входные данные:*

- $p, p \geq 2$  – заданное (исходное) число процессоров;
- $n, n \geq 2$  – число конкурирующих неоднородных распределённых процессов;
- $s, s \geq 2$  – число блоков линейно-структурированного программного ресурса;
- $M$  – двумерный массив, содержащий элементы исходной матрицы с учётом дополнительных системных расходов  $\varepsilon [t_{ij}^\varepsilon]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ ;
- $d$  – заданное (директивное) время выполнения конкурирующих процессов.

*Выходные данные:*

- $p^*$  – минимальное (оптимальное) число процессоров, обеспечивающих выполнение конкурирующих процессов за директивное время  $d$ ;
- $M_q$  – двумерный массив, содержащий результирующую матрицу времён выполнения блоков программного ресурса вида (1);

-  $q$  – порядковый номер результирующей матрицы времён выполнения блоков программного ресурса  $PR$ .

*Метод.*

Если  $d < T_n^{ac}(p^s, n, s, \varepsilon)$ , то полагаем  $p^* = 0$ , т.е. директивное время выполнения конкурирующих процессов  $d$  не может быть реализовано в заданных условиях ни для какого числа процессоров.

Пусть  $d \geq T_n^{ac}(p^s, n, s, \varepsilon)$  и число процессоров  $MC$  является ограниченным, т.е.  $s > p$ . Тогда между  $d$ ,  $T_n^{ac}(p^s, n, s, \varepsilon)$ ,  $T_n^{ac}(p < s, n, s, \varepsilon)$  возможны следующие случаи:

- если  $T_n^{ac}(p^s, n, s, \varepsilon) \leq d = T_n^{ac}(p, n, s, \varepsilon)$  или  $d > T_n^{ac}(p, n, s, \varepsilon)$ , то нахождение  $p^*$  осуществляется методом деления пополам отрезка  $[2, p]$ ;

- если  $T_n^{ac}(p^*, n, s, \varepsilon) \leq d < T_n^{ac}(p, n, s, \varepsilon)$ , то нахождение  $p^*$  осуществляется методом деления пополам отрезка  $[p, p^s]$ .

Пусть  $s \leq p$ . Тогда нахождение  $p^*$  осуществляется методом деления отрезка  $[2, p^s]$  пополам.

Нетрудно подсчитать, что сложность алгоритма нахождения оптимального числа процессоров  $p^*$ , базирующегося на предложенном методе, составляет величину  $O((k+1)np \log_2 p)$  операций в худшем случае.

На рис. 4 приводится графическая интерпретация зависимости величины  $T_n^{ac}(p, n, s, \varepsilon)$  от числа процессоров  $p$ , а также указаны величины  $d$ ,  $T_n^{ac}(p^s, n, s, \varepsilon)$ ,  $p^*$  и  $p^s$ . Из рисунка видно, что величина  $p^*$  определяется либо как точка пересечения прямой  $d$  с дискретной линией, определяющей зависимость  $T_n^{ac}(p, n, s, \varepsilon)$  от  $p$ , либо как ближайшая точка, которая находится ниже прямой  $d$ .

**Пример.** Пусть  $p = 3$ ,  $n = 3$ ,  $s = 9$ ,  $d = 48$ , а исходная матрица времён выполнения блоков с учётом дополнительных системных расходов по организации структуриро-

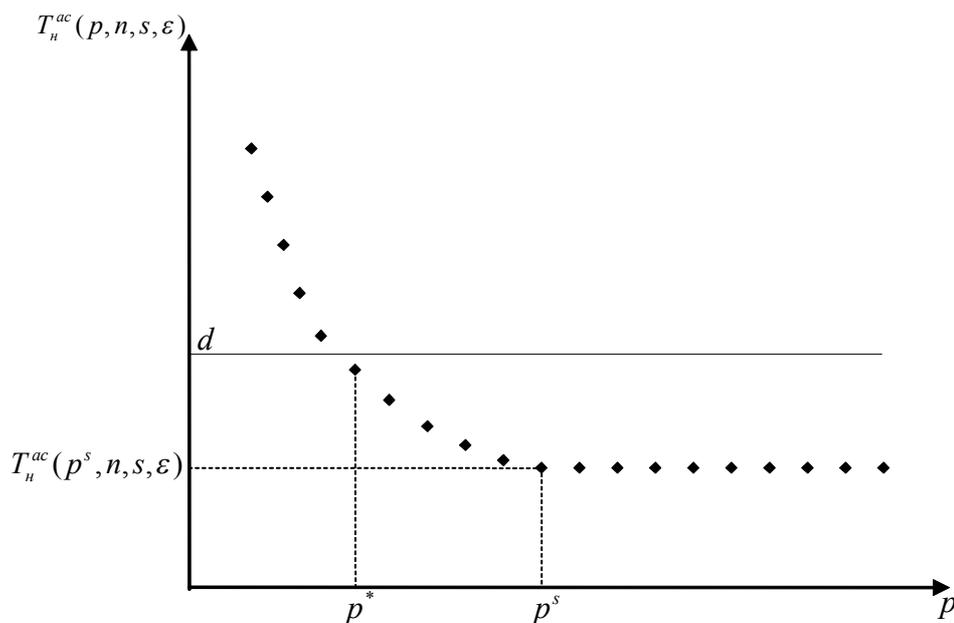


Рис. 4. Зависимости  $T_n^{ac}(p, n, s, \varepsilon)$  от числа процессоров

вания и параллельного использования блоков  $PR$   $\varepsilon$  имеет вид

$$T^\varepsilon = M = \begin{bmatrix} 4 & 1 & 3 & 5 & 2 & 4 & 7 & 3 & 1 \\ 2 & 6 & 4 & 1 & 5 & 3 & 4 & 2 & 8 \\ 5 & 3 & 1 & 7 & 4 & 2 & 6 & 4 & 5 \end{bmatrix}.$$

В данном случае достаточное число процессоров  $p^s = 9$ .

1. По исходной матрице  $M$  строим вершинно-взвешенный граф  $G_1^{ac}$  (рис. 5) и находим величину  $T_n^{ac}(p^s = 9, n, s, \varepsilon) = 45$ .

По исходным данным  $p, n, s$  и  $M$  строим результирующую матрицу  $T^*$  вида (1):

$$T^* = \begin{bmatrix} \begin{bmatrix} \hat{4} & 1 & 3 \\ \hat{2} & \hat{6} & \hat{4} \\ 5 & 3 & 1 \end{bmatrix} & \begin{bmatrix} 5 & 2 & 4 \\ \hat{1} & 5 & 3 \\ \hat{7} & 4 & 2 \end{bmatrix} & \begin{bmatrix} 7 & 3 & 1 \\ 4 & 2 & 8 \\ 6 & 4 & 5 \end{bmatrix} \\ \begin{bmatrix} 5 & 2 & 4 \\ 1 & 5 & 3 \\ 7 & 4 & 2 \end{bmatrix} & \begin{bmatrix} \hat{7} & 3 & 1 \\ \hat{4} & 2 & 8 \\ \hat{6} & \hat{4} & \hat{5} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 7 & 3 & 1 \\ 4 & 2 & 8 \\ 6 & 4 & 5 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{bmatrix}.$$

С помощью этой матрицы находим величину  $T_n^{ac}(p = 3, n, s, \varepsilon) = 50$ , которая и будет определять минимальное общее время выполнения неоднородных распределённых конкурирующих процессов в асинхронном режиме на  $p = 3$  процессорах. Оно совпадает со значением времени выполнения процессов в совмещенной диаграмме Ганта (рис. 6).

Учитывая, что

$$T_n^{ac}(p^s = 9, n, s, \varepsilon) = 45 \leq d = 48 < T_n^{ac}(p = 3, n, s, \varepsilon) = 50,$$

рассмотрим отрезок  $[3, 9]$ .

2. Делением отрезка  $[3, 9]$  пополам находим  $p_1 = 6$  и строим по заданным  $n, s$  и полученному  $p_1 = 6$  результирующую матрицу  $M_1$  вида (1):

$$M_1 = \begin{bmatrix} \begin{bmatrix} \hat{4} & 1 & 3 & 5 & 2 & 4 \\ \hat{2} & \hat{6} & \hat{4} & \hat{1} & 5 & 3 \\ 5 & 3 & 1 & \hat{7} & \hat{4} & \hat{2} \end{bmatrix} & \begin{bmatrix} 7 & 7 & 1 & 0 & 0 & 0 \\ 4 & 2 & 8 & 0 & 0 & 0 \\ \hat{6} & \hat{4} & \hat{5} & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 7 & 7 & 1 & 0 & 0 & 0 \\ 4 & 2 & 8 & 0 & 0 & 0 \\ 6 & 4 & 5 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix}.$$

С помощью матрицы  $M_1$  вычисляем величину  $T_n^{ac}(p_1 = 6, n, s, \varepsilon) = 45$ . Так как

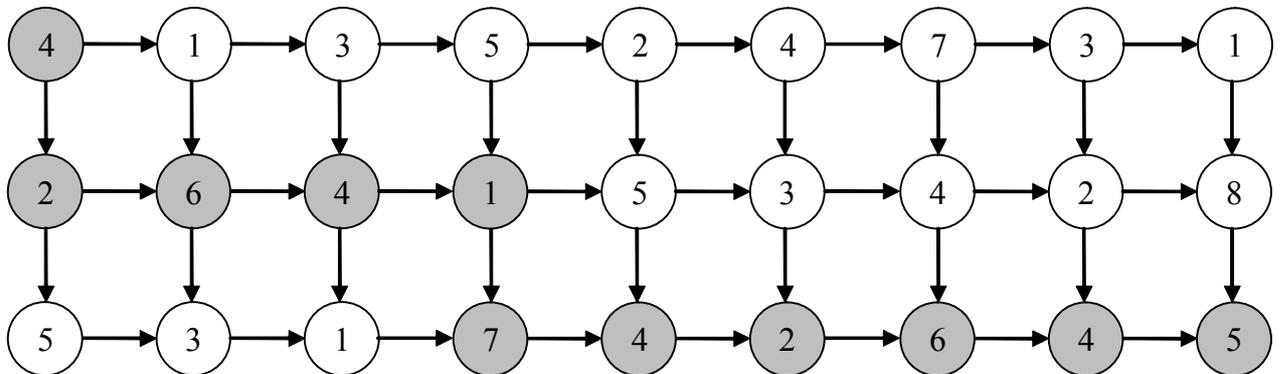


Рис. 5. Вершинно-взвешенный граф  $G_1^{ac}$

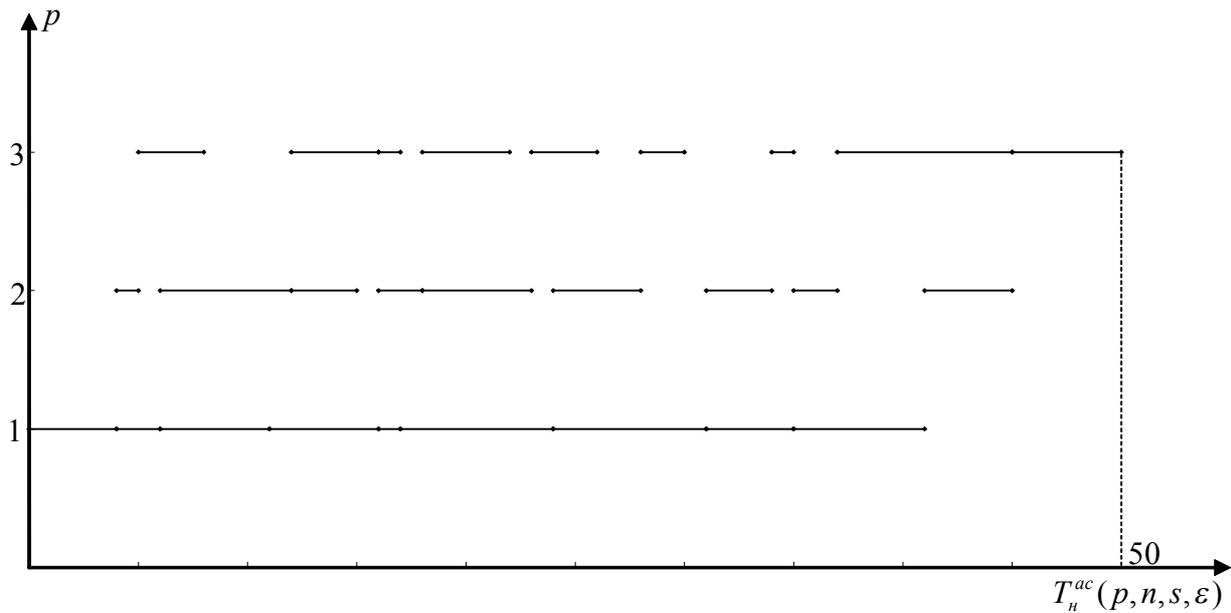


Рис. 6. Совмещенная диаграмма Ганта

$T_n^{ac}(p_1 = 6, n, s, \epsilon) = 45 \leq d = 48$ , то рассматриваем отрезок [3,6].

3. Находим величину  $p_2 = 4$ , которая является наименьшим целым полусуммы чисел 3 и 6. Далее по заданным  $n, s$  и полученному значению  $p_2 = 4$  строим результирующую матрицу  $M_2$  вида (1):

$$M_2 = \begin{bmatrix} \begin{bmatrix} \hat{4} & 1 & 3 & 5 \\ \hat{2} & \hat{6} & \hat{4} & \hat{1} \\ 5 & 3 & 1 & \hat{7} \end{bmatrix} & \begin{bmatrix} 2 & 4 & 7 & 7 \\ 5 & 3 & 4 & 2 \\ \hat{4} & \hat{2} & \hat{6} & \hat{4} \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ \hat{5} & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 2 & 4 & 7 & 7 \\ 5 & 3 & 4 & 2 \\ 4 & 2 & 6 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix}$$

С помощью матрицы  $M_2$  вычисляем величину  $T_n^{ac}(p_2 = 4, n, s, \epsilon) = 45$  (рис. 7). Таким образом, директивное время  $d = 48$  выполнения  $n = 3$  процессов реализуется при

$p_2 = 4$ , т.к.  $d = 48 > T_n^{ac}(p_2 = 4, n, s, \epsilon) = 45$ , и не реализуется при  $p = 3$ , т.к.  $d = 48 < T_n^{ac}(p = 3, n, s, \epsilon) = 50$ . Следовательно,  $p^* = 4$ .

**Заключение.** Предложенный метод нахождения минимального числа процессоров, обеспечивающих директивное время выполнения заданных объёмов вычислений в асинхронном режиме взаимодействия процессоров, процессов и блоков программного ресурса, является отправной точкой для решения подобных задач для первого и второго синхронного режимов. Результаты, полученные для одного структурированного  $PR$ , могут быть использованы при решении оптимизационных задач и в случае ограниченного числа программных ресурсов, что позволит установить взаимосвязи мультиконвейерной обработки с аналогичной обработкой при одном программном конвейере, провести математическое исследование эффективности и оптимальности мультиконвейерной организации конкурирующих процессов, вскрыть потенциальные возможности роста ускорения вычислений, выполнить сравнительный анализ различных режимов такой обработки.

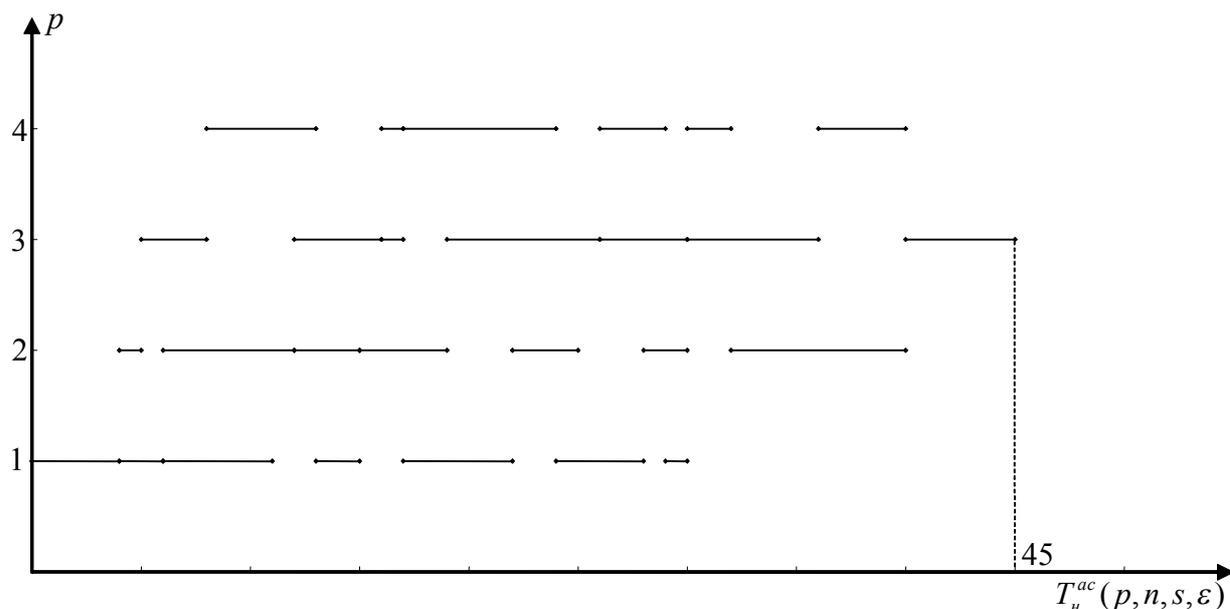


Рис. 7. Результирующая диаграмма Ганта

### Библиографический список

1. Воеводин, В. В. Параллельные вычисления [Текст]/ В. В. Воеводин, Вл. В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.
2. Коваленко, Н. С. Вычислительные методы реализации интеллектуальных моделей сложных систем [Текст]/ Н. С. Коваленко, С. А. Самаль. – Минск: Беларуская навука, 2004. – 166 с.
3. Иванников, В. П. О минимальном времени реализации распределенных конкурирующих процессов в синхронных режимах. [Текст]/ В. П. Иванников, Н. С. Коваленко, В. М. Метельский // Программирование. - 2000. - №5. - С. 44–52.
4. Капитонова, Ю. В. Оптимальность систем одинаково распределенных конкурирующих процессов [Текст]/ Ю. В. Капитонова, Н. С. Коваленко, П. А. Павлов // Кибернетика и системный анализ. - 2005. - № 6. - С. 3–10.
5. Павлов, П. А. Масштабируемые распределенные системы конкурирующих взаимодействующих процессов и их оптимальность [Текст]/ П. А. Павлов // Вестник Самарского государственного аэрокосмического университета. - 2010. - №1. - С. 234–243.

## PROBLEM OF OPTIMIZATION OF THE NUMBER OF PROCESSORS IN DISTRIBUTED PROCESSING

© 2011 P. A. Pavlov

Palesky State University

The problem of specifying time characteristics of distributed processes competing for a program resource is solved and a method of finding the optimum number of processors providing directive time of performing set volumes of calculations is suggested.

*Distributed process, program resource, asynchronous mode, homogeneous distributed system, unlimited parallelism, limited parallelism.*

**Информация об авторе**

**Павлов Павел Александрович**, кандидат физико-математических наук, доцент кафедры высшей математики и информационных технологий, Полесский государственный университет. E-mail: pin2535@tut.by. Область научных интересов: сосредоточенная, распределенная, макроконвейерная обработка параллельных вычислительных процессов.

**Pavlov Pavel Alexandrovitch**, candidate of physical and mathematical sciences, associate professor of the department of higher mathematics and information technologies, Palesky State University. E-mail: pin2535@tut.by. Area of research: concentrated, distributed processing of parallel computational processes.