УДК 681.3.06

Синхронные распределенные вычисления в условиях неограниченного параллелизма при ограниченном числе копий программного ресурса

П. А. Павлов
Полесский государственный университет, Беларусь

В статье решены задачи определения минимального общего времени выполнения распределенных конкурирующих процессов, при ограниченном числе копий программного ресурса в условиях неограниченного параллелизма.

Ключевые слова: *распределенный процесс, программный ресурс, асинхронный режим, ограниченный параллелизм*.

У статті вирішені питання визначення мінімального загального часу виповнення розподілених конкурентних процесів, при обмеженім числі копій програмного ресурсу в умовах необмеженого паралелізма.

Ключові слова*: расподілений процес, програмний ресурс, асинхронний режим, обмежений паралелізм*.

In article problems of definition of minimum general time of performance of the distributed competing processes are solved at the limited number of copies of a program resource in the conditions of unlimited parallelism.

Keywords: *distributed process, program resource, asynchronous mode, limited parallelism*.

1. Введение

Во многих приложениях, связанных с оптимальной организацией параллельных вычислений в многопроцессорных системах (МС) и вычислительных комплексах (ВК), значительный интерес представляют задачи, когда множество конкурирующих процессов могут использовать не одну, а несколько копий структурированного программного ресурса (ПР). Случай, когда в общей памяти многопроцессорной системы имеется одна копия ПР, с различных точек зрения был изучен в работах [1–11]. При этом были решены задачи определения минимального общего времени выполнения распределенных конкурирующих процессов, использующих структурированный на блоки программный ресурс в различных режимах взаимодействия процессов, процессоров и блоков [1–4], получены критерии эффективности и оптимальности структурирования программных ресурсов [5], проведен сравнительный анализ режимов взаимодействия процессов, процессоров и блоков [6–7], решен ряд оптимизационных задач по расчету числа процессов, минимального числа процессоров и др. [8–11]. Изучение этих и других задач, относящихся к оптимальной организации параллельных вычислений, приобретает особую актуальность в случае, когда в общей памяти МС могут быть одновременно размещены $c \geq 2$ копий программного ресурса. Такое обобщение носит принципиальный характер в виду того, что отражает основные черты мультиконвейерной обработки, а также позволяет сравнить эффективность конвейерной и параллельной обработки.

В данной работе строится и исследуется математическая модель организации конкурирующих процессов, использующих ограниченное число копий

© Павлов П.А., 2010

программного ресурса. При этом, используя идеи метода структурирования программных ресурсов на блоки с их последующей конвейеризацией по процессам и процессорам, исследуются оптимальные временные характеристики такой организации.

2. Математическая модель распределенных вычислений при ограниченном числе копий программного ресурса

Конструктивными элементами для построения математических моделей систем распределенных вычислений являются понятия процесса и программного ресурса [1–11].

Процесс будем рассматривать как последовательность блоков (команд, процедур) $Q_1, Q_2, ..., Q_s$, для выполнения которых используется множество процессоров (процессорных узлов, обрабатывающих устройств, интеллектуальных клиентов). При этом процесс называется *распределённым*, если все блоки или часть из них обрабатываются разными процессорами. Для ускорения выполнения процессы могут обрабатываться параллельно, взаимодействуя путем обмена информацией. Такие процессы называются *кооперативными* или *взаимодействующими* процессами.

Понятие *ресурса* используется для обозначения любых объектов вычислительной системы, которые могут быть использованы процессами для своего выполнения. *Реентерабельные (многократно используемые)* ресурсы характеризуются возможностью одновременного использования несколькими вычислительными процессами. Для параллельных систем характерной является ситуация, когда одну и ту же последовательность блоков или ее часть необходимо процессорам выполнять многократно, такую последовательность будем называть *программным ресурсом*, а множество соответствующих процессов – *конкурирующими*.

Математическая модель распределенной обработки конкурирующих взаимодействующих процессов при ограниченном числе копий программного ресурса включает в себя:

p, $p \ge 2$, процессоров многопроцессорной системы, которые имеют доступ к общей памяти;

n, $n \ge 2$, распределенных конкурирующих процессов;

s, $s \ge 2$, блоков структурированного на блоки программного ресурса;

матрицу $T = [t_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$, времен выполнения блоков программного ресурса распределенными взаимодействующими конкурирующими процессами;

$2 \le c \le p$, число копий структурированного на блоки программного ресурса, которые могут одновременно находиться в оперативной памяти, доступной для всех p процессоров, причем $\left[\frac{p}{c}\right] \ge 2$;

$\varepsilon > 0$ – параметр, характеризующий время дополнительных системных расходов, связанных с организацией конвейерного режима использования блоков структурированного программного ресурса множеством взаимодействующих конкурирующих процессов при распределенной обработке.

Будем также предполагать, что число процессов n кратно числу копий c структурированного программного ресурса, т.е. $n = mc$, $m \geq 2$, и что взаимодействие процессов, процессоров и блоков программного ресурса подчинено следующим условиям:

1) ни один из процессоров не может обрабатывать одновременно более одного блока;

2) процессы выполняются в параллельно–конвейерном режиме группами, т.е. осуществляется одновременное (параллельное) выполнение c копий каждого блока в сочетании с конвейеризацией групп из c блоков по процессорам и процессам;

3) обработка каждого блока программного ресурса осуществляется без прерываний;

4) распределение блоков программного ресурса по процессорам для каждого из процессов $i = lc + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, c}$, осуществляется циклически по правилу: блок с номером $j = k\left[\dfrac{p}{c}\right] + r$, $j = \overline{1, s}$, $k \geq 0$, $r = \overline{1, \left[\dfrac{p}{c}\right]}$, распределяется на процессор с номером $q + c(r-1)$.

Введем следующие режимы взаимодействия процессов, процессоров и блоков с учетом наличия c копий программного ресурса:

1) асинхронный режим, при котором начало выполнения очередной группы из c копий блока Q_j, $j = \overline{1, s}$, определяется наличием c процессоров и готовностью этой группы блоков к выполнению (программный блок считается готовым к выполнению, если он не выполняется ни на одном из процессоров);

2) первый синхронный режим, обеспечивающий линейный порядок выполнения блоков программного ресурса внутри каждого из процессов без задержек, т.е. для каждого процесса $i = lc + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, c}$, момент завершения выполнения j–го блока на $(q + c(r-1))$–м процессоре совпадает с моментом начала выполнения следующего $(j+1)$–го блока на $(q + cr)$–м процессоре, $j = \overline{1, s-1}$, $r = \overline{1, \left[\dfrac{p}{c}\right]}$;

3) второй синхронный режим, при котором c копий каждого блока непрерывно переходит по группам из c процессов, т.е. момент окончания обработки c копий текущего блока совпадает с моментом начала их обработки на следующей группе из c процессоров.

На Рис.1–3 представлены диаграммы Ганта, иллюстрирующие выполнение $n = 4$ распределенных конкурирующих процессов, использующих $c = 2$ копии структурированного программного ресурса в МС с $p = 7$ процессорами в рассмотренных выше режимах и с заданной матрицей времен выполнения блоков ПР с учетом дополнительных системных расходов $T^{\varepsilon} = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \end{bmatrix}$.

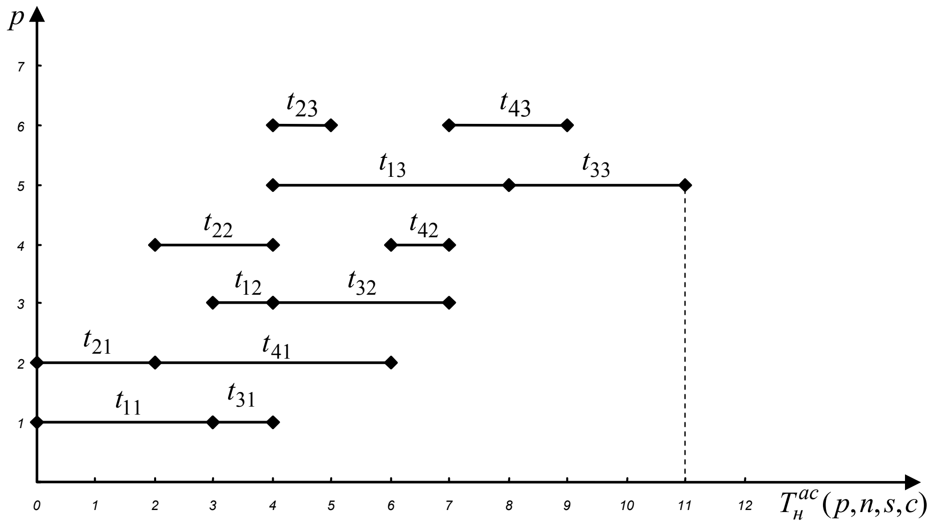

Рис.1. Асинхронный режим

Определение 1. Система n распределенных конкурирующих процессов называется *неоднородной*, если времена выполнения блоков программного ресурса $Q_1, Q_2, ..., Q_s$ зависят от объемов обрабатываемых данных и/или их структуры, т. е. разные для разных процессов.

Определение 2. Систему распределенных конкурирующих процессов будем называть *однородной*, если времена выполнения Q_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. Систему конкурирующих процессов будем называть *одинаково распределенной*, если времена t_{ij} выполнения блоков Q_j, $j = \overline{1,s}$, программного ресурса каждым из i–х процессов совпадают и равны t_i для всех $i = \overline{1,n}$, т.е. справедлива цепочка равенств $t_{i1} = t_{i2} = ... = t_{is} = t_i$ для всех $i = \overline{1,n}$.

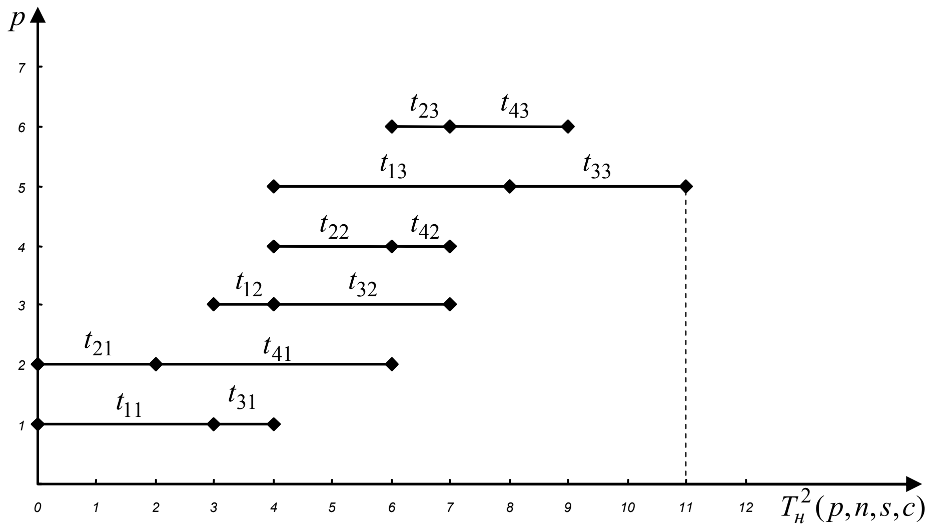

Рис. 2. Первый синхронный режим

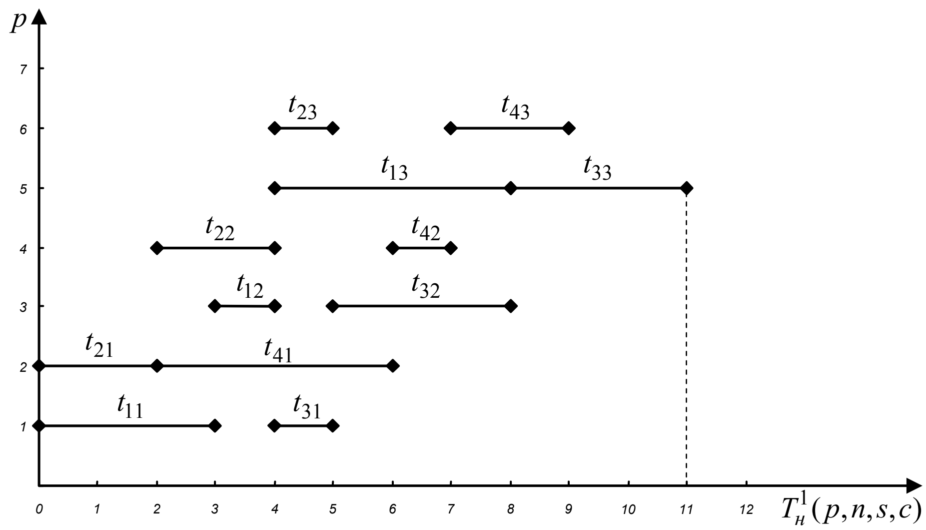

Рис.3. Второй синхронный режим

Заметим, что для системы однородных процессов в матрице времен выполнения блоков конкурирующими процессами будут одинаковы все строки, а для одинаково распределенной системы – столбцы.

3. Минимальное общее время выполнения неоднородных распределенных процессов в асинхронном режиме при достаточном числе процессоров

Обозначим минимальное общее время выполнения n неоднородных распределенных конкурирующих процессов при ограниченном числе c копий программного ресурса в многопроцессорной системе с p процессорами в асинхронном режиме, с учетом параметра ε, через $T_н^{ac}(p,n,s,c)$. Для вычисления $T_н^{ac}(p,n,s,c)$ рассмотрим случаи *неограниченного* $\left(s \le \left[\frac{p}{c}\right]\right)$ параллелизма.

Пусть имеется система $n = mc$, $m \ge 2$, $2 \le c \le p$, неоднородных распределенных конкурирующих процессов, причем число блоков s структурированного программного ресурса не превосходит числа групп процессоров по c процессоров в каждой, т.е. $2 \le s \le \left[\frac{p}{c}\right]$. В этом случае без ограничения общности можно считать, что каждый Q_j–й, $j = \overline{1,s}$, блок i–го процесса, где $i = lc + q$, $i = \overline{1,n}$, $l \ge 0$, $q = \overline{1,c}$, закреплен за $(q + c(r-1))$–м процессором, $r = \overline{1,\left[\frac{p}{c}\right]}$. Тогда для выполнения n процессов достаточно взять $p = \left[\frac{p}{c}\right]s$ процессоров, а остальные $p - \left[\frac{p}{c}\right]s$ процессоров будут не задействованы.

Пусть $T^\varepsilon = [t_{ij}^\varepsilon]$ – $n \times s$–матрица времен выполнения блоков программного ресурса каждым из i–х процессов с учетом параметра $\varepsilon > 0$, где $t_{ij}^\varepsilon = t_{ij} + \varepsilon$, $i = \overline{1,n}$, $j = \overline{1,s}$. Для вычисления минимального общего времени $T_н^{ac}(p,n,s,c)$ можно воспользоваться функционалом задачи Беллмана–Джонсона, который в нашем случае будет иметь вид:

$$T_н^{ac}(p,n,s,c) =$$

$$= \max_{1 \le u_1 \le u_2 \le \ldots \le u_{s-1} \le m} \left[\sum_{i=1}^{u_1} t_{q+(i-1)c,1}^\varepsilon + \sum_{i=u_1}^{u_2} t_{q+(i-1)c,2}^\varepsilon + \ldots + \sum_{i=u_{s-1}}^{m} t_{q+(i-1)c,s}^\varepsilon \right], \qquad (1)$$

где $m = \frac{n}{c}$, $t_{q+(i-1)c,j}^\varepsilon = t_{q+(i-1)c,j} + \varepsilon$, $q = \overline{1,c}$, $i = \overline{1,m}$, $j = \overline{1,s}$, а $u_1, u_2, \ldots, u_{s-1}$ – целые числа.

П р и м е р 1. Рассмотрим интерпретацию формулы (1) на числовом примере. Пусть $p = 7$, $n = 6$, $s = 3$, $c = 2$, а времена выполнения блоков процессами заданы матрицей $T^\varepsilon = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 1 \\ 1 & 3 & 3 \\ 4 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 4 & 1 \end{bmatrix}$. Тогда $m = 3$, следовательно, функционал (1) примет вид:

$$T_{н}^{ac}(p=7,n=6,s=3,c=2)=$$

$$=\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{q+(i-1)2,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{q+(i-1)2,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{q+(i-1)2,3}\right],\ q=\overline{1,2}\,.$$

1) При $q=1$, имеем:

$$\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{q+(i-1)2,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{q+(i-1)2,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{q+(i-1)2,3}\right]=$$

$$=\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{2i-1,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{2i-1,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{2i-1,3}\right].$$

Если $u_1=1$, $u_2=\overline{1,3}$, если $u_1=2$, $u_2=2,3$, если $u_1=3$, $u_2=3$, тогда:

$$\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{2i-1,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{2i-1,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{2i-1,3}\right]=$$

$$=\max\left[\begin{array}{ll}
\sum_{i=1}^{1} t^{\varepsilon}_{2i-1,1}+\sum_{i=1}^{1} t^{\varepsilon}_{2i-1,2}+\sum_{i=1}^{3} t^{\varepsilon}_{2i-1,3}, & \sum_{i=1}^{1} t^{\varepsilon}_{2i-1,1}+\sum_{i=1}^{2} t^{\varepsilon}_{2i-1,2}+\sum_{i=2}^{3} t^{\varepsilon}_{2i-1,3},\\
\sum_{i=1}^{1} t^{\varepsilon}_{2i-1,1}+\sum_{i=1}^{3} t^{\varepsilon}_{2i-1,2}+\sum_{i=3}^{3} t^{\varepsilon}_{2i-1,3}, & \sum_{i=1}^{2} t^{\varepsilon}_{2i-1,1}+\sum_{i=2}^{2} t^{\varepsilon}_{2i-1,2}+\sum_{i=2}^{3} t^{\varepsilon}_{2i-1,3},\\
\sum_{i=1}^{2} t^{\varepsilon}_{2i-1,1}+\sum_{i=2}^{3} t^{\varepsilon}_{2i-1,2}+\sum_{i=3}^{3} t^{\varepsilon}_{2i-1,3}, & \sum_{i=1}^{3} t^{\varepsilon}_{2i-1,1}+\sum_{i=3}^{3} t^{\varepsilon}_{2i-1,2}+\sum_{i=3}^{3} t^{\varepsilon}_{2i-1,3}
\end{array}\right]=$$

$$=\max\left[\begin{array}{ll}
\underline{t^{\varepsilon}_{1,1}+t^{\varepsilon}_{1,2}+t^{\varepsilon}_{1,3}+t^{\varepsilon}_{3,3}+t^{\varepsilon}_{5,3}}, & t^{\varepsilon}_{1,1}+t^{\varepsilon}_{1,2}+t^{\varepsilon}_{3,2}+t^{\varepsilon}_{3,3}+t^{\varepsilon}_{5,3},\\
t^{\varepsilon}_{1,1}+t^{\varepsilon}_{1,2}+t^{\varepsilon}_{3,2}+t^{\varepsilon}_{5,2}+t^{\varepsilon}_{5,3}, & t^{\varepsilon}_{1,1}+t^{\varepsilon}_{3,1}+t^{\varepsilon}_{3,2}+t^{\varepsilon}_{3,3}+t^{\varepsilon}_{5,3},\\
t^{\varepsilon}_{1,1}+t^{\varepsilon}_{3,1}+t^{\varepsilon}_{3,2}+t^{\varepsilon}_{5,2}+t^{\varepsilon}_{5,3}, & t^{\varepsilon}_{1,1}+t^{\varepsilon}_{3,1}+t^{\varepsilon}_{5,1}+t^{\varepsilon}_{5,2}+t^{\varepsilon}_{5,3}
\end{array}\right]=$$

$$=\max\left[\begin{array}{ll}
\underline{3+1+4+3+1}, & 3+1+3+3+1,\\
3+1+3+2+1, & 3+1+3+3+1,\\
3+1+3+2+1, & 3+1+3+2+1
\end{array}\right]=\max[12,11,10,11,10,10]=12\,.$$

2) При $q=2$, имеем:

$$\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{q+(i-1)2,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{q+(i-1)2,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{q+(i-1)2,3}\right]=$$

$$=\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{2i,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{2i,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{2i,3}\right].$$

Если $u_1=1$, $u_2=\overline{1,3}$, если $u_1=2$, $u_2=2,3$, $u_1=3$, $u_2=3$, тогда:

$$\max_{1\le u_1\le u_2\le 3}\left[\sum_{i=1}^{u_1} t^{\varepsilon}_{2i,1}+\sum_{i=u_1}^{u_2} t^{\varepsilon}_{2i,2}+\sum_{i=u_2}^{3} t^{\varepsilon}_{2i,3}\right]=$$

$$=\max\left[\begin{array}{ll}
\sum_{i=1}^{1} t^{\varepsilon}_{2i,1}+\sum_{i=1}^{1} t^{\varepsilon}_{2i,2}+\sum_{i=1}^{3} t^{\varepsilon}_{2i,3}, & \sum_{i=1}^{1} t^{\varepsilon}_{2i,1}+\sum_{i=1}^{2} t^{\varepsilon}_{2i,2}+\sum_{i=2}^{3} t^{\varepsilon}_{2i,3},\\
\sum_{i=1}^{1} t^{\varepsilon}_{2i,1}+\sum_{i=1}^{3} t^{\varepsilon}_{2i,2}+\sum_{i=3}^{3} t^{\varepsilon}_{2i,3}, & \sum_{i=1}^{2} t^{\varepsilon}_{2i,1}+\sum_{i=2}^{2} t^{\varepsilon}_{2i,2}+\sum_{i=2}^{3} t^{\varepsilon}_{2i,3},\\
\sum_{i=1}^{2} t^{\varepsilon}_{2i,1}+\sum_{i=2}^{3} t^{\varepsilon}_{2i,2}+\sum_{i=3}^{3} t^{\varepsilon}_{2i,3}, & \sum_{i=1}^{3} t^{\varepsilon}_{2i,1}+\sum_{i=3}^{3} t^{\varepsilon}_{2i,2}+\sum_{i=3}^{3} t^{\varepsilon}_{2i,3}
\end{array}\right]=$$

$$=\max\left[\begin{array}{ll}
t^{\varepsilon}_{2,1}+t^{\varepsilon}_{2,2}+t^{\varepsilon}_{2,3}+t^{\varepsilon}_{4,3}+t^{\varepsilon}_{6,3}, & t^{\varepsilon}_{2,1}+t^{\varepsilon}_{2,2}+t^{\varepsilon}_{4,2}+t^{\varepsilon}_{4,3}+t^{\varepsilon}_{6,3},\\
t^{\varepsilon}_{2,1}+t^{\varepsilon}_{2,2}+t^{\varepsilon}_{4,2}+t^{\varepsilon}_{6,2}+t^{\varepsilon}_{6,3}, & t^{\varepsilon}_{2,1}+t^{\varepsilon}_{4,1}+t^{\varepsilon}_{4,2}+t^{\varepsilon}_{4,3}+t^{\varepsilon}_{6,3},\\
\underline{t^{\varepsilon}_{2,1}+t^{\varepsilon}_{4,1}+t^{\varepsilon}_{4,2}+t^{\varepsilon}_{6,2}+t^{\varepsilon}_{6,3}}, & \underline{t^{\varepsilon}_{2,1}+t^{\varepsilon}_{4,1}+t^{\varepsilon}_{6,1}+t^{\varepsilon}_{6,2}+t^{\varepsilon}_{6,3}}
\end{array}\right]=$$

$$=\max\left[\begin{array}{ll}
2+2+1+2+1, & 2+2+1+2+1,\\
2+2+1+4+1, & 2+4+1+2+1,\\
\underline{2+4+1+4+1}, & \underline{2+4+1+4+1}
\end{array}\right]==\max[8,8,10,10,12,12]=12\,.$$

Следовательно, минимальное общее время выполнения $n=6$ неоднородных распределенных взаимодействующих конкурирующих процессов, использующих $c=2$ копии структурированного на $s=3$ блока программного ресурса, в многопроцессорной системе с $p=7$ процессорами в асинхронном режиме составит $T_{н}^{ac}(p,n,s,c)=12$. При этом будет использовано *6* процессоров.

Рассмотрим алгоритм, который позволяет решить задачу определения минимального общего времени $T_{н}^{ac}(p,n,s,c)$ выполнения неоднородных распределенных конкурирующих процессов в асинхронном режиме значительно эффективнее.

По заданным s, c, $m=\dfrac{n}{c}$ и матрице $T^{\varepsilon}=[t^{\varepsilon}_{q+(i-1)c,j}]$, $q=\overline{1,c}$, $i=\overline{1,m}$, $j=\overline{1,s}$, строим c–слойный вершинно–взвешенный граф G_1^c. Каждый q–й, $q=\overline{1,c}$, слой графа G_1^c состоит из вершин $t^{\varepsilon}_{q+(i-1)c,j}$, $i=\overline{1,m}$, $j=\overline{1,s}$, которые расположены в узлах прямоугольной $m\times s$–решетки, причем t^{ε}_{q1} – входные вершины, $t^{\varepsilon}_{q+t,s}$ – выходные, $q=\overline{1,c}$, $t=(m-1)c$ (Рис.4). Дуги в каждом слое q отражают линейный порядок выполнения блоков Q_j, $j=\overline{1,s}$, программного ресурса каждым из $(q+(i-1)c)$–м процессом, $q=\overline{1,c}$, $i=\overline{1,m}$, а также линейный порядок использования каждого блока всеми m процессами, Таким образом, имеет место следующая теорема.

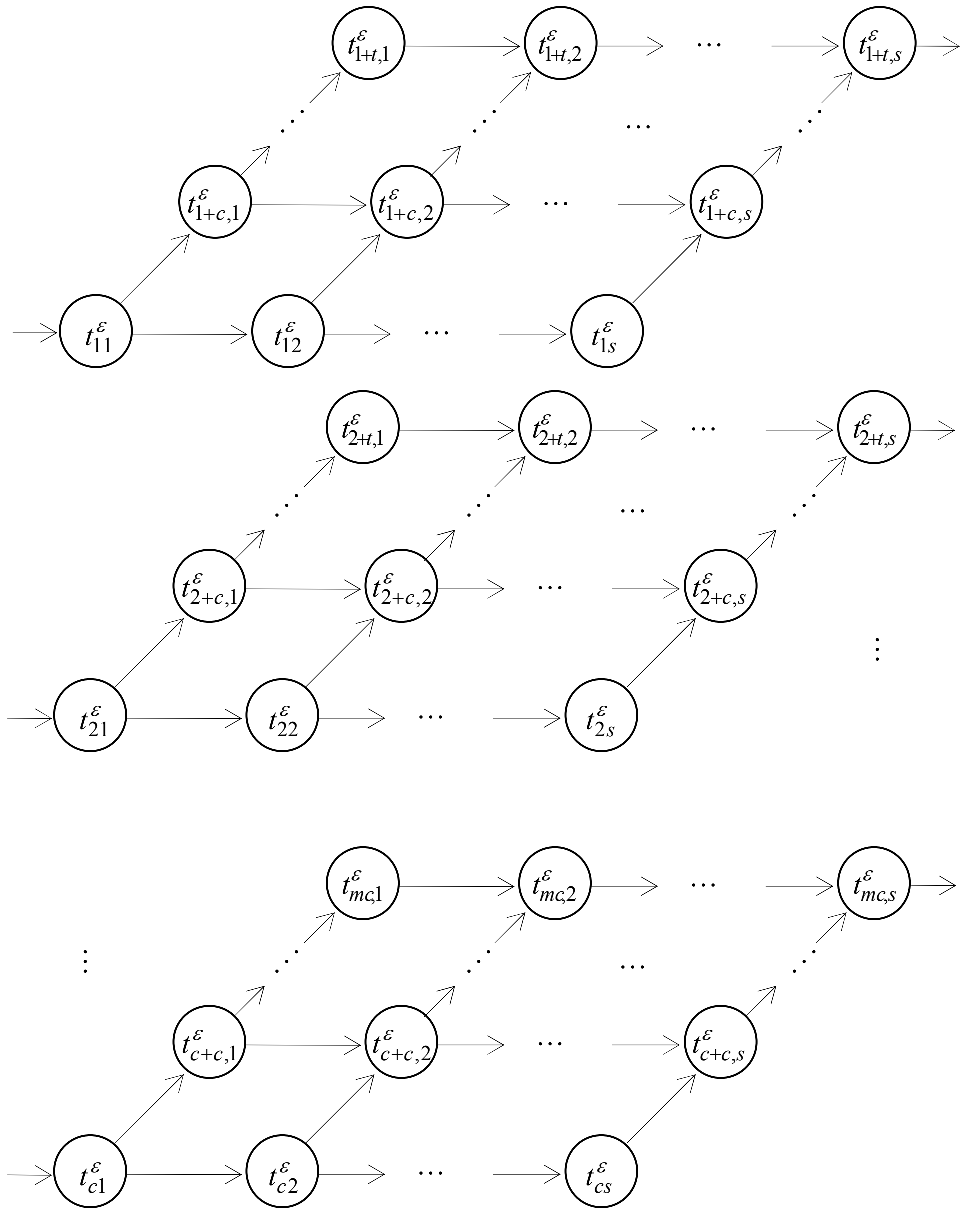

Рис.4. Граф G_1^c

Т е о р е м а 1. Минимальное общее время выполнения $n = mc$, $m \geq 2$, неоднородных распределенных конкурирующих процессов, использующих $2 \leq c \leq p$ копии структурированного на $2 \leq s \leq \left[\frac{p}{c}\right]$ блока программного ресурса с временами выполнения блоков с учетом дополнительных системных расходов $[t_{ij}^{\varepsilon}]$, $i=\overline{1,n}$, $j=\overline{1,s}$, в многопроцессорной системе с p, $\left[\frac{p}{c}\right] \geq 2$, процессорами в асинхронном режиме определяется длиной критического пути в c–слойном вершинно–взвешенном графе G_1^c из начальной вершины t_{q1}^{ε} в конечную $t_{q+(m-1)c,s}^{\varepsilon}$, $q=\overline{1,c}$, $m=\frac{n}{c}$.

П р и м е р 2. Используя данные примера 1, найти минимальное общее время $T_н^{ac}(p,n,s,c)$, используя алгоритм нахождения критического пути в *c*–слойном вершинно–взвешенном графе G_1^c.

По заданным $n=6$, $s=3$, и матрице T^{ε} строим 2–слойный ($c=2$) вершинно–взвешенный граф G_1^c (Рис.5). Каждый слой содержит *ms* вершин, где $m=n/c=3$. Длина критического пути в графе равна *12*, что совпадает с минимальным общим временем, полученном в примере 1.

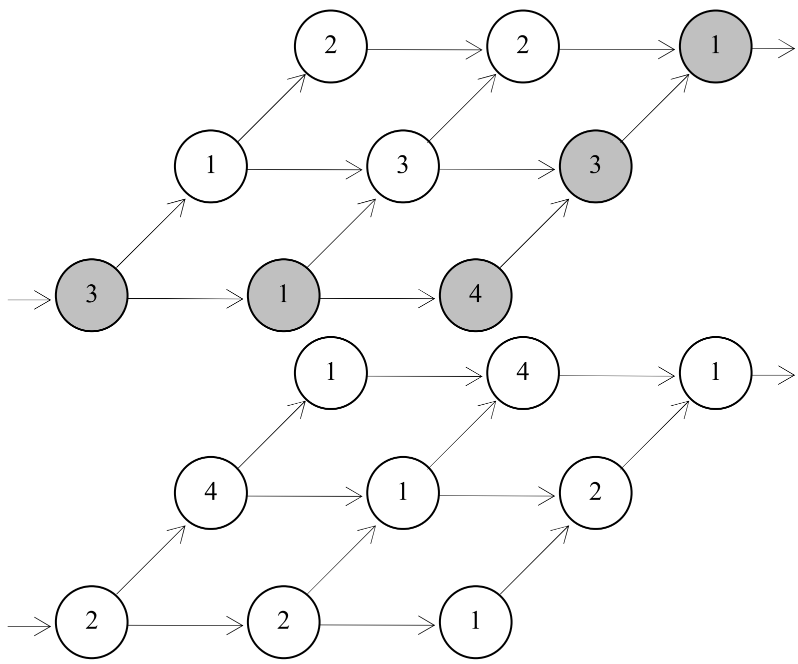

Рис.5. C–слойный вершинно–взвешенный граф G_1^c

4. Время выполнения однородных и одинаково распределенных конкурирующих процессов

Согласно определению 2 систему распределенных конкурирующих процессов будем считать *однородной*, если времена выполнения каждого блока Q_j, $j=\overline{1,s}$, каждым из процессов равны, т.е. $t_{ij}^{\varepsilon}=t_j^{\varepsilon}$, $i=\overline{1,n}$, $j=\overline{1,s}$.

На Рис.6 представлена диаграмма Ганта, иллюстрирующая выполнение однородных распределенных конкурирующих процессов при ограниченном числе копий программного ресурса в МС с параметрами: $p=7$, $n=4$, $s=3$, $c=2$, $T^{\varepsilon}=\begin{bmatrix}3&1&4\\3&1&4\\3&1&4\\3&1&4\end{bmatrix}$.

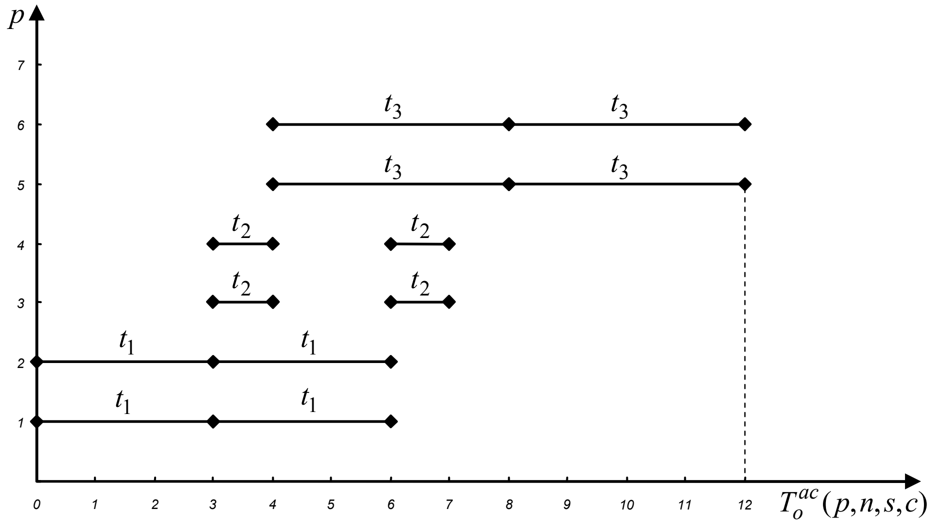

Рис.6. Асинхронный режим – однородные процессы

Оценим общее время выполнения n однородных распределенных конкурирующих процессов в асинхронном режиме, использующих с копий структурированного программного ресурса. Пусть $(t_1^{\varepsilon},t_2^{\varepsilon},...,t_s^{\varepsilon})$ – длительности выполнения каждого из блоков Q_j, $j=\overline{1,s}$, программного ресурса с учетом накладных расходов ε, $t_j^{\varepsilon}=t_j+\varepsilon$, $j=\overline{1,s}$. Обозначим длительность выполнения всего программного ресурса каждым из процессов через

$$T_{\varepsilon}^{s}=\sum_{j=1}^{s}t_j^{s}.$$

Покажем, что в этих условиях вычисление общего времени $T_o^{ac}(p,n,s,c)$ в случае неограниченного параллелизма сводится к нахождению общего времени выполнения однородных распределенных процессов при одной копии структурированного программного ресурса. При $n=mc$, $m\geq 2$, $2\leq c\leq p$, выполнение c копий структурированного программного ресурса в асинхронном режиме равносильно выполнению c групп по m процессов, конкурирующих за использование одной копии программного ресурса на $\left[\dfrac{p}{c}\right]$ процессорах.

В силу формулы вычисления общего времени выполнения n однородных конкурирующих процессов, использующих одну копию структурированного программного ресурса, полученной в [1,4], и с учетом того, что $n = mc$, $m \geq 2$, $2 \leq c \leq p$, получаем:

$$T_o^{ac}(p, mc, s, c) = T_o^{ac}\left(\left[\frac{p}{c}\right], m, s, 1\right) = T_\varepsilon^s + (m-1)\max_{1 \leq j \leq s} t_j^\varepsilon .$$

Матрица времен выполнения блоков программного ресурса в этом случае будет иметь размерность $n \times s$ и состоять из n одинаковых строк.

Рассмотрим систему *одинаково распределенных* конкурирующих процессов. Времена выполнения всех блоков рассматриваемой системы с учетом накладных расходов ε каждым из i–х процессов совпадают и равны t_i^ε, т. е. справедлива цепочка равенств $t_{i1}^\varepsilon = t_{i2}^\varepsilon = ... = t_{is}^\varepsilon = t_i^\varepsilon$, для всех $i = \overline{1, n}$.

На Рис.7 представлена диаграмма Ганта выполнения одинаково распределенных конкурирующих процессов в МС с параметрами: $p = 7$, $n = 4$, $s = 3$, $c = 2$, $T^\varepsilon = \begin{bmatrix} 3 & 3 & 3 \\ 4 & 4 & 4 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$, в случае *неограниченного* $\left(s \leq \left[\dfrac{p}{c}\right]\right)$ параллелизма.

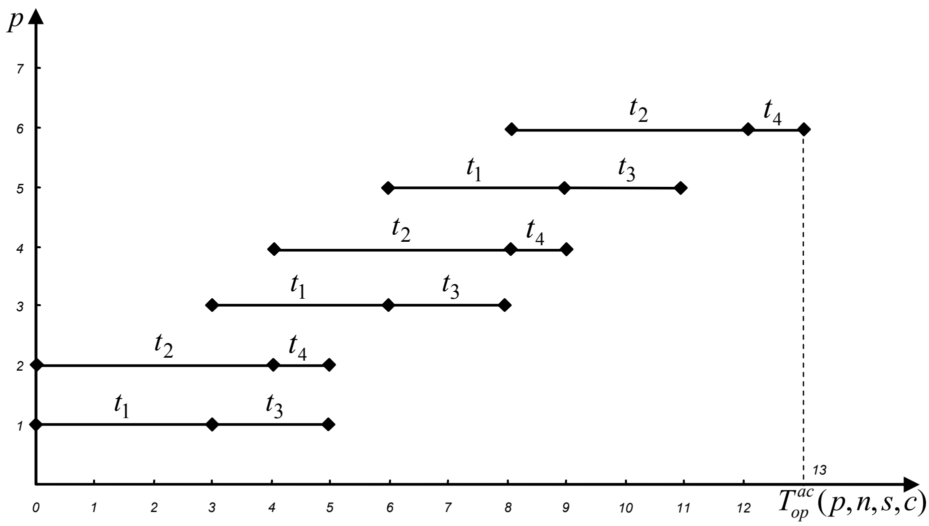

Рис.7. Асинхронный режим – одинаково распределенные процессы

Обозначим через $T_{\varepsilon}^{q}=\sum_{i=1}^{m} t_{q+(i-1)c}^{\varepsilon}$ – суммарное время выполнения каждого из блоков Q_j, $j=\overline{1,s}$, всеми m процессами из q–й группы, а $t_{\max}^{q}=\max_{1\le i\le m} t_{q+(i-1)c}^{\varepsilon}$ – максимальное время выполнения блока из этой группы, $q=\overline{1,c}$. Справедлива следующая теорема.

Т е о р е м а 3. *Минимальное общее время выполнения n, $n \ge 2$, одинаково распределенных конкурирующих процессов, использующих структурированный на s, $s \ge 2$, блоков программный ресурс в многопроцессорной системе с p, $p \ge 2$, процессорами в асинхронном режиме при ограниченном числе копий программного ресурса составляет величину $T_{op}^{ac}(p,n,s,c)$, равную*

$$T_{op}^{ac}(p,n,s,c)=\max_{1\le q\le c}\left(T_{\varepsilon}^{q}+(s-1)t_{\max}^{q}\right).$$

Для доказательства воспользуемся функционалом (1) задачи Беллмана–Джонсона, который для системы m одинаково распределенных конкурирующих процессов из q–й группы примет вид:

$$T_{н}^{ac}(p,n,s,c)=\max_{1\le u_1\le u_2\le\ldots\le u_{s-1}\le m}\left[\sum_{i=1}^{u_1} t_{q+(i-1)c}^{\varepsilon}+\sum_{i=u_1}^{u_2} t_{q+(i-1)c}^{\varepsilon}+\ldots+\sum_{i=u_{s-1}}^{m} t_{q+(i-1)c}^{\varepsilon}\right]=$$

$$=T_{\varepsilon}^{q}+(s-1)t_{\max}^{q},$$

где $m=\frac{n}{c}$, $t_{q+(i-1)c}^{\varepsilon}=t_{q+(i-1)c}+\varepsilon$, $q=\overline{1,c}$, $i=\overline{1,m}$, u_1, u_2, ..., u_{s-1} – целые числа.

4. Заключение

В заключении отметим, что рассмотренное обобщение математической модели с одним программным ресурсом (конвейером) на случай ограниченного числа программных ресурсов позволяет установить взаимосвязи мультиконвейерной обработки с аналогичной обработкой при одном программном конвейере, получить аналитические оценки общего времени выполнения конкурирующих процессов при ограниченном параллелизме и провести математическое исследование эффективности и оптимальности мультиконвейерной организации конкурирующих процессов, вскрыть потенциальные возможности роста ускорения вычислений, выполнить сравнительный анализ различных режимов такой обработки.

ЛИТЕРАТУРА

1. Коваленко Н.С., Метельский В.М. О времени реализации конкурирующих процессов при распределенной обработке. // Кибернетика и системный анализ. – 1996. – №1. – С. 54–64.

2. Коваленко Н.С., Метельский В.М. Режимы взаимодействия неоднородных распределенных конкурирующих процессов // Кибернетика и системный анализ. – 1997. – №3. – С. 31–43.
3. Иванников В.П., Коваленко Н.С., Метельский В.М. О минимальном времени реализации распределенных конкурирующих процессов в синхронных режимах // Программирование. – 2000. – №5. – С. 44–52.
4. Павлов П.А. О времени реализации систем параллельных распределенных процессов // Вестник Полесского государственного университета. – 2008. – №1. – С. 60–67.
5. Коваленко Н.С., Павлов П.А. Эффективность и оптимальность структурирования программных ресурсов при распределенной обработке // Труды минского института управления. – 2005. – №1. – С. 104–107.
6. Павлов П.А. Анализ режимов организации одинаково распределенных конкурирующих процессов // Вестник БГУ. Сер. 1: Физ.Мат.Информ. – 2006. – №1. – С. 116–120.
7. Павлов П.А. Сравнительный анализ одинаково распределенных конкурирующих процессов с учетом дополнительных системных расходов // Вестник Фонда фундаментальных исследований. – 2006. – №1. – С. 55–58.
8. Капитонова Ю.В., Коваленко Н.С., Павлов П.А. Оптимальность систем одинаково распределенных конкурирующих процессов // Кибернетика и системный анализ. – 2005. – №6. – С. 3–10.
9. Коваленко Н.С., Павлов П.А. Эффективность систем конкурирующих процессов с учетом накладных расходов // Доклады НАН Беларусі. Сер. фіз.–мат. навук. – 2005. – №6. – С. 32–36.
10. Коваленко Н.С., Павлов П.А. Системы одинаково распределенных конкурирующих процессов в условиях ограниченного параллелизма и их оптимальность // Доклады НАН Беларусі. Сер. фіз.–мат. навук. – 2006. – №2. – С. 25–29.
11. Yu.V. Kapitonova, N.S. Kovalenko, P.A. Pavlov, Optimality of systems of identically distributed competing processes, Cybernetics and Systems Analysis. – New York: Springer, 2006. – P. 793–799.

Надійшла - 28.10.2010.