

## О ВРЕМЕНИ РЕАЛИЗАЦИИ СИСТЕМ ПАРАЛЛЕЛЬНЫХ РАСПРЕДЕЛЕННЫХ ПРОЦЕССОВ

**П.А. ПАВЛОВ**

*Полесский государственный университет  
г. Пинск, Республика Беларусь, pin2535@tut.by*

### ВВЕДЕНИЕ

Постоянное существование задач сверхвысокой сложности (проблемы экономики, экологии, космических исследований, изучения биологических и химических процессов, материаловедения и др.), которые нужно решать в указанных сферах, характеризуются большой размерностью, десятками сотен и миллионов независимых переменных и соответствующих ограничений. Подобные задачи актуальны и для Республики Беларусь. Это, прежде всего, задачи оптимизационного плана развития экономики страны или отдельного региона, задачи проектирования сложных систем (самолетов, энергетических котлов, ракетной техники), сооружений, технологических процессов, задачи использования спутников Земли для развития народного хозяйства, задачи военного характера. Указанные задачи можно эффективно решать, используя распараллеливание сложных процессов и обработки больших объемов данных и знаний, применяя математические методы и модели.

### 1. КОНСТРУКТИВНЫЕ ЭЛЕМЕНТЫ ПАРАЛЛЕЛЬНЫХ СИСТЕМ

Основой для построения математических моделей параллельных систем являются понятия процесса и программного ресурса [1].

*Процесс* будем рассматривать как последовательность блоков (команд, наборов действий), для выполнения которых используется множество интеллектуальных клиентов (процессоров, обрабатывающих устройств, роботов). При этом процесс называется *распределенным*, если все блоки или часть из них обрабатываются разными интеллектуальными клиентами.

Моменты времени начала выполнения каждого блока определяются последовательностью  $(t_1, t_2, \dots, t_s)$ . Предполагая, что блоки выполняются строго последовательно, в ходе своей реализации являются неделимыми и имеют длительности выполнения  $d_j > 0$ ,  $j = \overline{1, s}$ , получим, что моменты времени завершения выполнения блоков определяются последовательностью вида  $(t_1 + d_1, t_2 + d_2, \dots, t_s + d_s)$ . Процесс является *активным* и находится в состоянии *выполнения*, если достигается равенство  $t_{j+1} = t_j + d_j$ ,  $j = \overline{1, s-1}$ , т. е. для выполнения процесса выделено обрабатывающее устройство, и после завершения выполнения очередного блока процесса сразу же начинается выполнение следующего блока. Соотношение  $t_{j+1} > t_j + d_j$ ,  $j = \overline{1, s-1}$ , означает, что после выполнения очередного блока процесс *приостановлен* и ожидает возможности для продолжения своего выполнения. Данная приостановка может быть вызвана необходимостью разделения использования интеллектуального клиента между одновременно исполняемыми процессами. В этом случае приостановленный процесс находится в состоянии *ожидания* момента предоставления обрабатывающего устройства для своего выполнения. Кроме того, приостановка процесса может быть вызвана и временной неготовностью процесса к дальнейшему выполнению. В подобных ситуациях говорят, что процесс является *блокированным* и находится в состоянии *блокировки*.

Для ускорения выполнения процессы могут обрабатываться параллельно разными обрабатывающими устройствами, влияя на поведение друг друга путем обмена информацией. Такие процессы называются *кооперативными* или *взаимодействующими* процессами.

Итак, *параллельную систему распределенных процессов можно рассматривать как некоторый агрегированный процесс, получаемый путем параллельного объединения составляющих взаимодействующих процессов*.

Понятие *ресурса* используется для обозначения любых объектов параллельной системы, которые могут быть использованы процессами для своего выполнения. По характеру использования могут различаться следующие категории ресурсов [2]:

- *выделяемые* (монопольно используемые, неперераспределяемые) ресурсы характеризуются тем, что выделяются процессам в момент их возникновения и освобождаются только в момент завершения процессов;
- *повторно распределяемые* ресурсы отличаются возможностью динамического запрашивания, выделения и освобождения в ходе выполнения процессов;

- *разделяемые* ресурсы, особенность которых состоит в том, что они постоянно остаются в общем использовании и выделяются процессам для использования в режиме разделения времени;
- *реентерабельные* (многократно используемые) ресурсы характеризуются возможностью одновременного использования несколькими процессами.

Для параллельных систем характерной является ситуация, когда одну и ту же последовательность команд (действий, блоков) или ее часть необходимо интеллектуальным клиентам выполнять многократно, которую будем называть *программным ресурсом*, а множество соответствующих процессов – *конкурирующими*.

Решая проблему распределения программных ресурсов как следствие неявно решаются задачи эффективного использования остальных категорий ресурсов. С другой стороны, эффективно решая задачу распределения программных ресурсов, решаем проблему сокращения времени реализации множества параллельных распределенных процессов.

## 2. МЕТОД СТРУКТУРИРОВАНИЯ

*Структурирование (декомпозиция)* – это основной способ уменьшения времени выполнения больших задач, вычислений и вообще различных проблем [1]. Структурирование всегда предполагает то или иное разбиение задачи на части с последующей организацией линейного или частичного порядка на множестве этих частей. Часто вообще неясно, как правильно осуществить разбиение процесса или задачи, хотя это наиболее важная и наиболее трудная область решений.

Пусть  $PR$  – программный ресурс,  $n, n \geq 2$ , число конкурирующих процессов за его использование. *Требуется организовать процесс решения задачи таким образом, чтобы общее время выполнения процессов, использующих программный ресурс, было минимальным.* Одной из стратегий решения данной задачи с  $p, p \geq 2$  интеллектуальными клиентами является предоставление для каждого процесса отдельной копии  $PR$ . Но этот путь не всегда осуществим из-за ограниченного объема ресурсов и тем более трудно достижим в случае больших ресурсов, используемых в качестве многократно используемых. Поэтому при решении данной задачи применяется стратегия последовательного предоставления программного ресурса  $n$  процессам. В этом случае суммарное время выполнения  $n$  процессов составит величину  $T_{sum} = nT$ , где  $T$  – время выполнения каждым из процессов  $PR$ .

Время  $T_{sum}$  можно существенно сократить, если обеспечить структурирование реентерабельного программного ресурса на блоки  $Q_1, Q_2, \dots, Q_s$  с последующей конвейеризацией как блоков по процессам, так и по интеллектуальным клиентам распределенной системы. Для этого необходимо выполнить следующие основные этапы:

- структурировать программный ресурс на блоки  $Q_1, Q_2, \dots, Q_s$ ;
- организовать параллельное (одновременное) взаимодействие процессов, блоков и обрабатываемых устройств;
- согласовать во времени выполнение различных процессов, использующих одну и ту же копию каждого блока  $Q_j, j = \overline{1, s}$ ;
- запомнить после завершения использования очередного блока и восстановить перед началом выполнения следующего блока промежуточных состояний процессов;
- запустить процессы на выполнение и завершить их;
- выбрать режим (способ) взаимодействия процессов, устройств и блоков;
- обеспечить «командную» поддержку взаимодействия процессов, обрабатываемых устройств и блоков.

Структурирование программного ресурса на блоки осуществляется, как правило, либо исходя из физического смысла задачи на этапах создания математической модели и алгоритмов ее решения, либо путем анализа последовательного процесса с целью его декомпозиции. Число блоков, на которое осуществляется структурирование программного ресурса, зависит от количества процессов и обрабатываемых устройств, длительности выполнения процесса, накладных расходов и других параметров.

Один из возможных механизмов (способов) взаимодействия процессов, интеллектуальных клиентов и блоков следующий. Блоки, процессы и интеллектуальные клиенты нумеруются в порядке  $1, 2, \dots, s, 1, 2, \dots, n$  и  $1, 2, \dots, p$  соответственно. Причем на множестве блоков установлен линейный порядок их выполнения. Предполагается, что все  $n$  процессов используют одну и ту же копию структурированного программного ресурса. В дальнейшем под процессом будем понимать выполнение всех блоков в порядке  $1, 2, \dots, s$ .

Специально выделенный организующий процесс предоставляет блоки структурированного  $PR$   $Q_1, Q_2, \dots, Q_s$  каждому из процессов в порядке  $1, 2, \dots, n$ . Если блок  $Q_j, j = \overline{1, s}$ , освобождается очередным  $i$ -м процессом, то он предоставляется  $(i+1)$ -му процессу, а сам  $i$ -й процесс получает в свое распоряжение  $(j+1)$ -й блок, либо переводится в состояние ожидания до освобождения  $(j+1)$ -го блока,  $i = \overline{1, n-1}, j = \overline{1, s-1}$  и т. д. В случае распределенной обработки монополизация обрабатываемых устройств процессами не происходит, а блоки одного и того же процесса выполняются на разных устройствах.

Очевидно, что при наличии в распределенной системе  $p$  интеллектуальных клиентов возможно совмещенное во времени выполнение процессов. Запоминание и восстановление промежуточных состояний процессов, запуск процессов на выполнение и их завершение, набор режимов взаимодействия процессов, обрабатываемых устройств и блоков осуществляет специальный организующий процесс.

Проблема обоснования метода структурирования программных ресурсов на параллельно выполняемые блоки носит комплексный характер, что порождает ряд сложных в математическом отношении задач. Для их решения требуется прежде всего построение математических моделей, адекватно отражающих различные аспекты взаимодействия множества процессов, с учетом их физической специфики, дополнительных затрат и т. д. Анализ показывает, что на пути решения этой комплексной проблемы возникают математические задачи дискретно-комбинаторного характера, для решения которых эффективны аппарат теории графов, линейные диаграммы Ганта, теория расписаний, методы комбинаторной оптимизации.

### 3. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПАРАЛЛЕЛЬНЫХ РАСПРЕДЕЛЕННЫХ ПРОЦЕССОВ

Будем говорить, что нижеперечисленные объекты образуют математическую модель конкурирующих распределенных процессов:  $p$ ,  $p \geq 2$ , интеллектуальных клиентов;  $n$ ,  $n \geq 2$ , распределенных конкурирующих процессов;  $s$ ,  $s \geq 2$ , блоков структурированного программного процесса; матрицу  $T = [t_{ij}]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ , времен выполнения блоков процессами. Введем в рассмотрение параметр  $\varepsilon > 0$ , характеризующий накладные расходы, затрачиваемые на организацию параллельного выполнения блоков программного ресурса множеством конкурирующих процессов [1, 3].

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

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

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  $t_{i1} = t_{i2} = \dots = t_{is} = t_i$  для всех  $i = \overline{1, n}$ .

### 4. РЕЖИМЫ ОРГАНИЗАЦИИ ПРОЦЕССОВ, ИНТЕЛЛЕКТУАЛЬНЫХ КЛИЕНТОВ И БЛОКОВ ПРОГРАММНОГО РЕСУРСА

Предполагается, что взаимодействие обрабатываемых устройств, процессов и блоков программного ресурса подчинено следующим условиям:

- 1) ни один из блоков программного ресурса не может обрабатываться одновременно более чем одним интеллектуальным клиентом;
- 2) ни один из интеллектуальных клиентов не может обрабатывать одновременно более одного блока;
- 3) обработка каждого блока программного ресурса осуществляется без прерываний;
- 4) распределение блоков программного ресурса по обрабатываемым устройствам 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 обрабатывающее устройство с номером  $i$ .

Введем дополнительное условие, которое определяет *асинхронный режим* взаимодействия процессов, интеллектуальных клиентов и блоков:

- 1) отсутствуют простои обрабатываемых устройств при условии готовности блоков, а также невыполнение блоков при наличии устройств.
- 5 Введем также условия:
- 2) для каждого из  $n$  процессов момент завершения выполнения  $j$ -го блока  $i$ -м интеллектуальным клиентом совпадает с моментом начала выполнения следующего  $(j+1)$ -го блока на  $(i+1)$ -м обрабатываемом устройстве,  $i = \overline{1, p-1}$ ,  $j = \overline{1, s-1}$ ;
  - 3) для каждого из блоков момент завершения его выполнения  $l$ -м устройством совпадает с моментом начала его выполнения  $(l+1)$ -м процессом на том же обрабатываемом устройстве,  $l = \overline{1, n-1}$ .

Если к условиям 1–4 добавить поочередно условия 6 и 7 соответственно, то получим два базовых синхронных режима:

- *первый синхронный режим*, определяемый условиями 1–4, 6, обеспечивает непрерывное выполнение блоков программного ресурса внутри каждого из процессов;

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

## 5. ВРЕМЯ РЕАЛИЗАЦИИ РАСПРЕДЕЛЕННЫХ ПРОЦЕССОВ

Интерес представляют задачи, связанные с получением математических соотношений минимального общего времени взаимодействия обрабатывающих устройств, процессов и блоков структурированного программного ресурса в различных режимах.

*Асинхронный режим.*

Обозначим минимальное общее время выполнения  $n$  неоднородных распределенных процессов на  $p$  обрабатывающих устройствах в асинхронном режиме, с учетом введенного параметра  $\varepsilon$ , через  $T_n^{ac}(p, n, s, \varepsilon)$ . Для нахождения  $T_n^{ac}(p, n, s, \varepsilon)$  рассмотрим случаи *неограниченного* ( $2 \leq s \leq p$ ) и *ограниченного* ( $s > p$ ) параллелизма.

Пусть число блоков структурированного программного ресурса не превосходит числа интеллектуальных клиентов, т. е.  $2 \leq s \leq p$ . В этом случае без ограничения общности можно считать, что каждый  $Q_j$ -й блок закреплен за  $j$ -м обрабатывающим устройством,  $J = \overline{1, s}$ . Тогда для выполнения  $n$  процессов достаточно взять  $p = s$  обрабатывающих устройств, а остальные  $p - s$  будут не задействованы. Поэтому для определения минимального общего времени  $T_n^{ac}(p, n, s, \varepsilon)$  можно воспользоваться функционалом задачи Беллмана-Джонсона, который в нашем случае будет иметь вид [4]:

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

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 числа.

Задачу определения  $T_n^{ac}(p, n, s, \varepsilon)$  можно решить более эффективно с помощью аппарата сетевых вершинно-взвешенных графов (рис.1) [5]. Минимальное общее время в случае неограниченного параллелизма будет определяться длиной критического пути из начальной вершины в конечную. Каждой вершине графа соответствует значение  $t_j^\varepsilon = t_j + \varepsilon$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ .

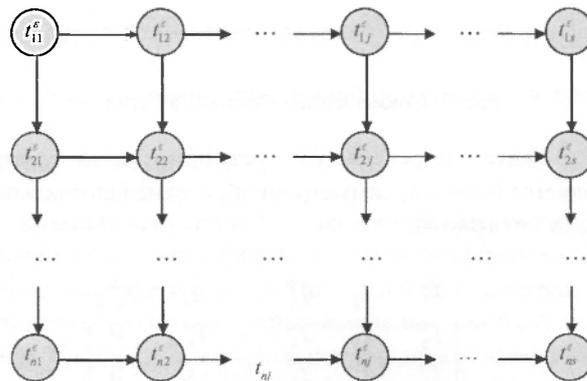


Рис. 1. Сетевой вершинно-взвешенный граф

В случае ограниченного параллелизма, т.е. когда  $s > p$ ,  $s = kp + r$ ,  $k \geq 1$ ,  $1 \leq r < p$ , исходную матрицу времен выполнения блоков процессами с учетом дополнительных системных расходов  $T^\varepsilon = [t_{ij}^\varepsilon]$ ,  $j = \overline{1, kp + r}$  разбиваем на  $(k+1)$ -ую подматрицу  $T_l^\varepsilon$ ,  $l = \overline{1, k+1}$  размерностью  $n \times p$  каждая. По каждой из подматриц строим линейную диаграмму Ганта, которые отражают во времени обработку очередных  $p$  блоков всех  $n$  процессов  $p$  обрабатывающими устройствами ( $p=3$ ,  $s=8$ ). Последняя диаграмма отражает выполнение последних  $r=2$  блоков (рис.2).

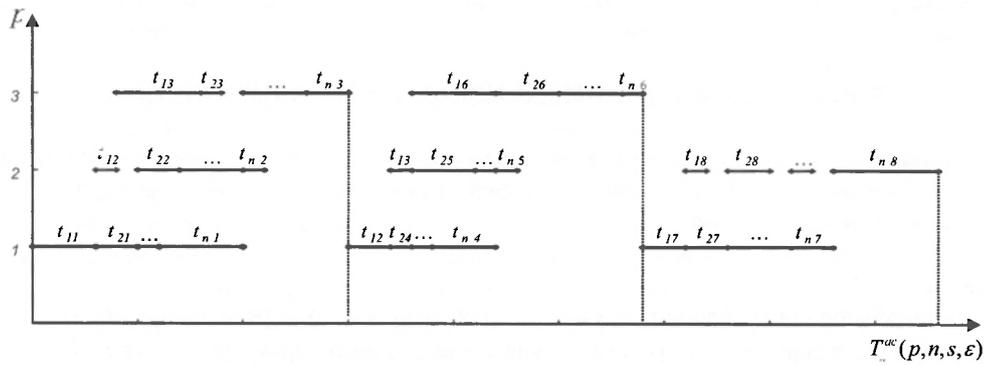


Рис. 2. Несовмещенная диаграмма Ганта

В этом случае общее время выполнения всех процессов будет определяться как сумма длин критических путей каждой из подряд идущих несовмещенных диаграмм Ганта. Это время можно существенно сократить, если воспользоваться приемом совмещения последовательных диаграмм Ганта по оси времени справа налево. Причем совмещение осуществляется поблочно, начиная со второй диаграммы, на максимально возможную величину таким образом, чтобы не нарушались условия, определяющие асинхронный режим. В результате совмещения получим результирующую совмещенную диаграмму Ганта (рис.3).

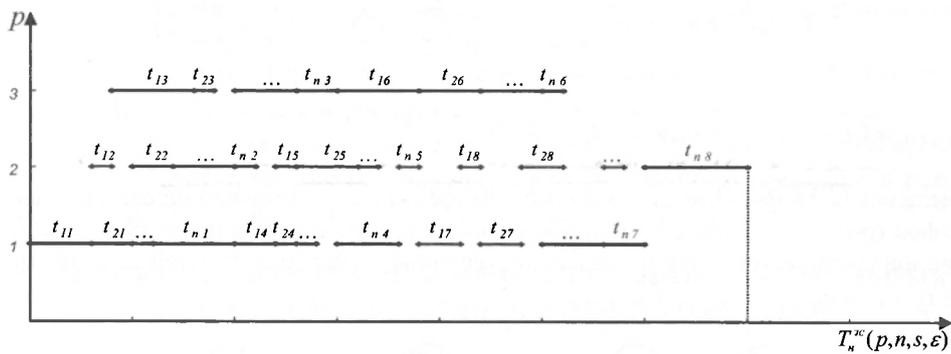


Рис. 3. Совмещенная диаграмма Ганта

Совмещенная диаграмма Ганта определяется результирующей матрицей  $T^*$  размерностью  $(k+1)n \times (k+1)p$ , которая является блочной, симметричной, верхнедиагональной относительно второй диагонали, типа Ганкелевой порядка  $k+1$  (рис.4).

$$T^* = \begin{pmatrix} T_1^\epsilon & T_2^\epsilon & T_3^\epsilon & \dots & T_k^\epsilon & T_{k+1}^\epsilon \\ T_2^\epsilon & T_3^\epsilon & T_4^\epsilon & \dots & T_{k+1}^\epsilon & 0 \\ T_3^\epsilon & T_4^\epsilon & T_5^\epsilon & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ T_k^\epsilon & T_{k+1}^\epsilon & 0 & \dots & 0 & 0 \\ T_{k+1}^\epsilon & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Рис. 4. Матрица  $T^*$

Для однородной системы конкурирующих процессов, т.е. когда времена выполнения  $Q_j$ -го блока каждым из процессов равны, т.е.  $t_{ij}^\epsilon = t_j^\epsilon$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, s}$ , минимальное общее время в случае  $2 \leq s \leq p$  в асинхронном режиме составляет величину  $T_o^{ac}(p, n, s, \epsilon)$ , равную:

$$T_o^{ac}(p, n, s, \epsilon) = T_\epsilon^s + (n-1) \max_{1 < j < s} t_j^\epsilon,$$

где  $(t_1^\varepsilon, t_2^\varepsilon, \dots, t_s^\varepsilon)$  – длительности выполнения каждого из блоков с учетом накладных расходов,

а  $T_\varepsilon^s = \sum_{j=1}^s t_j^\varepsilon$  – длительность выполнения всего распределенного процесса.

Для системы *одинаково распределенных* конкурирующих процессов, когда справедлива цепочка равенств  $t_{i1}^\varepsilon = t_{i2}^\varepsilon = \dots = t_{is}^\varepsilon = t_i^\varepsilon$ ,  $t_i^\varepsilon = t_i + \varepsilon$ , для всех  $i = \overline{1, n}$ , минимальное общее время составляет величину  $T_p^{ac}(p, n, s, \varepsilon)$ , равную

$$T_{op}^{ac}(p, n, s, \varepsilon) := \begin{cases} T_\varepsilon^n + (s-1) \max_{1 < i < n} t_i^\varepsilon, & s \leq p, \\ kT_\varepsilon^n + (p-1) \max_{1 < i < n} t_i^\varepsilon, & s = kp, k > 1, \\ (k+1)T_\varepsilon^n + (r-1) \max_{1 < i < n} t_i^\varepsilon, & s = kp + r, k \geq 1, 1 \leq r < p, \end{cases}$$

где  $T_\varepsilon^n = \sum_{i=1}^n t_i^\varepsilon$  – суммарное время выполнения каждого из блоков  $Q_j$ ,  $j = \overline{1, s}$ , всеми  $n$  конкурирующими процессами.

*Первый синхронный режим* [3].

В случае, когда число интеллектуальных клиентов является *достаточным*, т. е.  $s \leq p$ , для определения величины минимального общего времени выполнения *неоднородных* взаимодействующих процессов в первом синхронном режиме  $T_n^1(p, n, s, \varepsilon)$  имеет место формула:

$$T_n^1(p, n, s, \varepsilon) = \sum_{i=1}^{n-1} \max_{1 \leq u \leq s} \left[ \sum_{j=1}^u t_{ij}^\varepsilon - \sum_{j=1}^{u-1} t_{i+1,j}^\varepsilon \right] + \sum_{j=1}^s t_{nj}^\varepsilon,$$

где  $\max_{1 < u < p} \left[ \sum_{j=1}^u t_{ij}^\varepsilon - \sum_{j=1}^{u-1} t_{i+1,j}^\varepsilon \right]$ ,  $i = \overline{1, n-1}$  – моменты времени начала выполнения очередного процесса по отношению к предыдущему,  $\sum_{j=1}^p t_{nj}^\varepsilon$  – время выполнения последнего процесса. Эту формулу можно получить также из функционала многостадийной задачи теории расписаний при условии, что каждое требование обслуживается непрерывно одним прибором за другим [4].

В случае, когда число обрабатываемых устройств является *недостаточным*, например,  $s = kp$ ,  $k > 1$ , аналогично с асинхронным режимом выполняется разбиение матрицы времен выполнения блоков программного ресурса  $T^\varepsilon$  на  $k$  подматриц по  $p$  столбцов в каждой. По каждой из подматриц строится линейная диаграмма Ганта. Воспользовавшись приемом совмещения последовательных диаграмм по оси времени, для вычисления минимального общего времени  $T_n^1(p, n, kp, \varepsilon)$  в первом синхронном режиме имеет место выражение

$$T_n^1(p, n, kp, \varepsilon) = \sum_{l=1}^k T_l^\varepsilon - \sum_{l=1}^{k-1} \delta_l,$$

где  $T_l^\varepsilon = \sum_{i=1}^{n-1} \max_{1 \leq u \leq p} \left[ \sum_{j=1}^u t_{ij}^{\varepsilon, l} - \sum_{j=1}^{u-1} t_{i+1,j}^{\varepsilon, l} \right] + \sum_{j=1}^p t_{nj}^{\varepsilon, l}$ ,  $l = \overline{1, k}$  – время выполнения  $l$ -й группы блоков программного ресурса всеми  $n$  процессами  $p$  интеллектуальными клиентами, а  $\delta_l = \min\{\delta_l, \delta_l\}$  – длина отрезка максимально возможного совмещения двух последовательных диаграмм Ганта по оси времени. В свою очередь,  $\delta_l$  – разность между моментом начала выполнения  $j$ -го блока первого процесса в  $(l+1)$ -й группе блоков и моментом завершения выполнения  $j$ -го блока последней процесса в  $l$ -й группе блоков:

$$\delta_l = \min_{1 < j < p} \left[ \sum_{w=j+1}^p t_{nw}^{\varepsilon, l} + \sum_{w=1}^{j-1} t_{1w}^{\varepsilon, l+1} \right], \quad l = \overline{1, k-1};$$

$\delta_l^*$  – разность между началом выполнения первого блока  $i$ -м процессом в  $(l+1)$ -й группе блоков и моментом завершения выполнения  $p$ -го блока  $i$ -м процессом в  $l$ -й группы блоков:

$$\delta_l^* = \min_{1 \leq i < n} [T_i^\varepsilon + E_{i1}^{\varepsilon, l+1} - t_{i1}^{\varepsilon, l+1} - E_{ip}^{\varepsilon, i}], \quad l = \overline{1, k-1},$$

где  $t_{ij}^{\varepsilon, l} = t_{ij}^i + \varepsilon = t_{i, (l-1)p+j} + \varepsilon$  – время выполнения  $i$ -м процессом  $j$ -го блока в  $l$ -й группе блоков с учетом параметра  $\varepsilon$ , а  $E_{ij}^{\varepsilon, l} = \sum_{q=1}^{l-1} \max_{1 \leq u < p} \left[ \sum_{w=1}^u t_{aw}^{\varepsilon, l} - \sum_{w=1}^{u-1} t_{a+1, w}^{\varepsilon, l} \right] + \sum_{w=1}^i t_{iw}^{\varepsilon, l}$  – время завершения обработки  $i$ -м процессом  $j$ -го блока программного ресурса в  $l$ -й группе блоков,  $i = \overline{1, n}$ ,  $j = \overline{1, p}$ ,  $l = \overline{1, k}$ .

Если взаимодействие обрабатываемых устройств, процессов и блоков структурированного программного ресурса осуществляется в первом базовом синхронном режиме, то минимальное общее время  $T_o^1(p, n, s, \varepsilon)$  выполнения *однородных* распределенных конкурирующих процессов с учетом параметра накладных расходов  $\varepsilon > 0$  определяется следующим образом:

$$T_o^1(p, n, s, \varepsilon) = \begin{cases} (n-1) \max_{1 \leq j \leq s} t_j^\varepsilon + \sum_{j=1}^s t_j^\varepsilon, & s \leq p, \\ \sum_{l=1}^k T_l^\varepsilon - \sum_{l=1}^{k-1} \min\{\phi_l, \phi_l^*\}, & s = kp, \quad k > 1, \\ \sum_{l=1}^k T_l^\varepsilon + T_{k+1}^\varepsilon - \sum_{l=1}^{k-1} \phi_l - \phi_k, & s = kp + r, \quad k \geq 1, \quad 1 \leq r < p, \end{cases}$$

где

$$\phi_l^* = \min_{1 \leq j < p} \left[ \sum_{w=j+1}^p t_w^{\varepsilon, l} + \sum_{w=1}^{j-1} t_w^{\varepsilon, l+1} \right], \quad \phi_l^* = (n-1) \min \left[ \max_{1 \leq j \leq p} t_j^{\varepsilon, l+1}, \max_{1 \leq j \leq p} t_j^{\varepsilon, l} \right], \quad l = \overline{1, k-1},$$

$$T_{k+1}^\varepsilon = (n-1) \max_{1 \leq j \leq r} t_j^{\varepsilon, k+1} + \sum_{j=1}^r t_j^{\varepsilon, k+1},$$

$$\phi_k = \min\{\phi_k^*, \phi_k^*\},$$

$$\phi_k^* = \min_{1 \leq j < r} \left[ \sum_{w=1}^{j-1} t_w^{\varepsilon, k+1} + \sum_{w=j+1}^p t_w^{\varepsilon, k} \right], \quad \phi_k^* = (n-1) \min \left[ \max_{1 \leq j < r} t_j^{\varepsilon, k+1}, \max_{1 \leq j < p} t_j^{\varepsilon, k} \right].$$

Если взаимодействие обрабатываемых устройств, процессов и блоков структурированного программного ресурса подчинено условиям первого базового синхронного режима, то для любых  $p \geq 2$ ,  $n \geq 2$ ,  $s \geq 2$ ,  $\varepsilon > 0$  минимальное общее время выполнения *одинаково распределенных* конкурирующих процессов определяется по формулам:

$$T_{op}^1(p, n, s, \varepsilon) = \begin{cases} \sum_{i=1}^n t_i^\varepsilon + (s-1) \left[ t_n^\varepsilon + \sum_{i=2}^n \max\{t_{i-1}^\varepsilon - t_i^\varepsilon, 0\} \right], & s \leq p, \\ kT_{op}^1(p, n, p, \varepsilon) - (k-1) \min\{\omega_1, \omega_2\}, & s = kp, \quad k > 1, \\ kT_{op}^1(p, n, p, \varepsilon) + T_{op}^1(p, n, r, \varepsilon) - (k-1) \min\{\omega_1, \omega_2\} - \min\{\xi_1, \xi_2\}, \\ s = kp + r, \quad k \geq 1, \quad 1 \leq r < p. \end{cases}$$

где

$$T_{op}^1(p, n, p, \varepsilon) = \sum_{i=1}^n t_i^\varepsilon + (p-1) \left[ t_n^\varepsilon + \sum_{i=2}^n \max\{t_{i-1}^\varepsilon - t_i^\varepsilon, 0\} \right],$$

$$\omega_1 = (p-1) \min\{t_1^\varepsilon, t_n^\varepsilon\}, \quad \omega_2 = T_{op}^1(p, n, p, \varepsilon) - p \max_{1 \leq i < n} t_i^\varepsilon,$$

$$T_{op}^1(p, n, r, \varepsilon) = \sum_{i=1}^n t_i^\varepsilon + (r-1) \left[ t_n^\varepsilon + \sum_{i=2}^n \max\{t_{i-1}^\varepsilon - t_i^\varepsilon, 0\} \right],$$

$$\xi_1 = (r-1) \min\{t_1^\varepsilon, t_n^\varepsilon\} + (p-r)t_n^\varepsilon,$$

$$\xi_2 = \bar{t}_m^1(p, n, p, \varepsilon) - \max_{1 \leq i \leq n} [T_{op}^1(p, i, p, \varepsilon) - T_{op}^1(p, i, r, \varepsilon) + rt_i].$$

## ВЫВОДЫ

Решены задачи определения минимального общего времени реализации множества распределенных кооперативных процессов, конкурирующих за многократно используемый ресурс. Решение было проведено для систем параллельных распределенных процессов для асинхронного и первого синхронного режимов взаимодействия процессов, интеллектуальных клиентов и блоков структурированного программного ресурса. При этом был использован аппарат теории расписаний и теории сетевых графов, что позволило разработать методы нахождения точных значений минимального общего времени выполнения распределенных процессов в параллельных распределенных системах. Концепция построения параллельных распределенных систем конкурирующих процессов и их математическая основа является базой для эффективной организации процессов различной природы в экономике, медицине, технологических процессах и т.д.

## ЛИТЕРАТУРА

1. Коваленко, Н.С. Вычислительные методы реализации интеллектуальных моделей сложных систем / Н.С. Коваленко, С.А. Самаль. – Мн.: Бел. наука, 2004. – 166 с.
2. Гергель, В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем / В.П. Гергель, Р.Г. Стронгин. – Н. Новгород: ННГУ, 2001. – 122 с.
3. Kapitonova, Yu. V. Optimality of systems of identically distributed competing processes / Yu. V. Kapitonova, N.S. Kovalenko, P.A. Pavlov // Cybernetics and Systems Analysis. – New York: Springer, 2006. – P. 793–799.
4. Танаев, В.С. Теория расписаний. Многостадийные системы / В.С. Танаев, Ю.Н. Сотсков, В.А. Струсевич. – М.: Наука, 1989. – 328 с.
5. Емеличев, В.А. Лекции по теории графов / В.А. Емеличев, О.И. Мельников, В.И. Сарванов, Р.И. Тышкевич. – М.: Наука, 1990. – 384 с.
6. Коваленко, Н.С. Математическая модель организации производственного процесса на поточных линиях / Н.С. Коваленко, П.А. Павлов // Проблемы прогнозирования и государственного регулирования социально-экономического развития: Материалы VII Междунар. науч. конф., Минск, 19–20 окт. 2006 г. – Мн.: НИЭИ Министерства экономики Республики Беларусь, 2007. – С. 326–336.

## ABOUT THE TIME OF REALIZATION OF SYSTEMS OF THE PARALLEL DISTRIBUTED PROCESSES

*P.A. PAVLOV.*

### *Summary*

The author has solved the problems of definition the minimal general time necessary for realization a set of the distributed cooperative processes. The decision is led for asynchronous and the first synchronous modes of interaction of processes, intellectual clients and blocks of the structured programme resource.

*Поступила в редакцию 12 июня 2008 г.*