

Тема 3

Исследование функционирования МВС с различной организацией при решении сложных задач, моделируемых взвешенным оргграфом, и анализ производительности в зависимости от структуры МВС и стратегии решения задач.

1.ВВЕДЕНИЕ.....	1
2.ОПРЕДЕЛЕНИЕ КРИТИЧЕСКОГО ПУТИ НА ГРАФЕ ЗАДАЧИ.....	2
3.ПОСТАНОВКА ЗАДАЧИ НАЗНАЧЕНИЯ.....	10

1.Введение

Цикл лабораторных работ посвящен одному из основных разделов курса, связанному с организацией параллельных вычислений в многопроцессорных вычислительных системах (МВС). Рассматриваемые системы относятся к классу "множественный поток команд, множественный поток данных" (МКМД). Основными компонентами современных МВС являются:

- **решающее поле**, состоящее из однородных или разнородных процессоров, со своей локальной памятью (или без нее);
- **коммутационная сеть**, позволяющая вести обмен данными как между процессорами, так и между процессорами и общей памятью;
- **общая память модульного типа**;
- **иерархическая система управления**, позволяющая реализовать параллелизм обработки данных на уровне независимых задач, независимых ветвей задач, команд и операций.

В качестве параметров компонент системы используются:

- **решающее поле** – число процессоров (в данном цикле лабораторных работ рассматриваются МВС с одинаковыми процессорами), характеризующихся относительным временем выполнения независимых ветвей задачи, определяемым быстродействием процессоров;
- **локальная память (ЛП) процессора** - емкость (предполагается, что быстродействие ЛП таково, что не вносит задержку при обращении к ней процессора за данными, которые в ней находятся);
- **коммутационная сеть (КС)** – определяется типом (общая шина или мультиплексная шина), числом шин, пропускной способностью при передаче данных (характеризуется относительным временем занятия

сформированного канала: процессор - процессор, процессор - общая память, общая память – процессор);

- **общая память** - число модулей памяти, быстродействие, косвенно отображаемое временем занятия канала процессор - общая память, общая память - процессор.

Затраты на управление в моделях не учитываются.

На вход такой МВС в общем случае может поступать набор независимых входных задач, каждая из которых имеет свой приоритет.

В качестве модели задачи используется направленный граф, узлы которого отображают подзадачи. Каждой подзадаче соответствует часть (ветвь) программы, начав выполнение которой процессор заканчивает ее без прерываний. Дуги графа моделируют связи по данным между соответствующими подзадачами (ветвями программы).

Узлы графа взвешиваются целыми числами, соответствующими временам их выполнения в условных единицах (например, тактах) на процессорах. Дуги графа взвешиваются тоже целыми числами, соответствующими временам занятия канала связи (шины) при передаче данных от одного узла графа к другому.

В настоящем цикле лабораторных работ рассматриваются графы задач без обратных связей, с различным соотношением времен выполнения узлов графа (t_p) и передачи данных (t_n). В соответствии с этим соотношением различают **слабосвязанные задачи** ($t_p \gg t_n$), **среднесвязанные** ($t_p \approx t_n$), **сильносвязанные** ($t_p \ll t_n$).

Каждая задача может характеризоваться максимальным временем выполнения T_{\max} и минимальным временем T_{\min} . Например, для задач, у которых для всех узлов $t_n = 0$, T_{\max} вычисляется как сумма времен выполнения всех узлов графа (следует иметь в виду, что это справедливо, если все узлы графа выполняются на одном и том же процессоре), а T_{\min} - как сумма времен выполнения узлов графа, принадлежащих его критическому пути. Рассмотрим алгоритм поиска критического пути графа, реализованный в программных моделях для данного цикла лабораторных работ.

2.Определение критического пути на графе задачи

Суть алгоритма заключается в **определении минимально возможного и максимально возможного времени начала выполнения узлов графа**. В разработанных моделях используется следующий алгоритм

определения критических путей (КП) в графе со взвешенными узлами и дугами, то есть с учетом времен обработки узлов графа задачи и передачи данных между узлами графа. Данный алгоритм, аналогично соответствующему алгоритму для графа без учета времени передачи по магистрали, основан на последовательном проходе по дереву графа от начальной вершины (вершин) к конечной и возврата к начальной вершине. При начальном проходе (рис. 1) определяется минимально возможное время начала выполнения каждого узла по формуле:

$$T_{\min i} = \max_{j=1}^s (T_{\min j} + t_j + T_{ij}) \quad (1)$$

где s - число узлов-предшественников i -го узла;

$T_{\min i}(j)$ - минимально возможное время начала выполнения i -го (j -го) узла;

t_j - время выполнения j -го узла;

T_{ji} - время передачи данных между узлами j и i , которому присваивается одно из значений множества $\{0, \tau_{ji}, 2\tau_{ji}\}$ в зависимости от способа организации памяти, где τ_{ji} - время передачи данных между узлами j и i , задаваемое на исходном графе задачи.

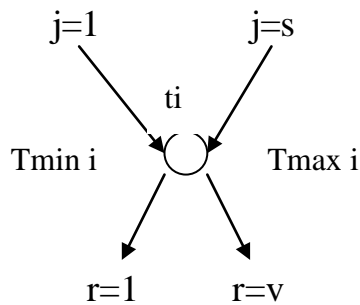


Рис. 1. Пояснения к определению критических узлов

При повторном анализе графа при проходе от конечной вершины к начальной определяется максимально возможное время начала выполнения узла по формуле

$$T_{\max i} = \min_{r=1}^v (T_{\max r} - t_i - T_{ir}) \quad (2)$$

где v - число узлов-последователей i -го узла;

$T_{\max i}(r)$ - максимально возможное время начала выполнения i -го (r -го) узла;

t_i - время выполнения i -го узла;

T_{ir} - время передачи данных i -го узла узлу r , значение которому присваивается аналогично T_{ij}

При этом для начальной вершины (вершин) $T_{\min i} = 0$, а для конечной вершины минимально возможное время начала ее выполнения совпадает с максимально возможным временем начала выполнения - $T_{\max i} = T_{\min i}$.

i -й узел является критическим, если выполняется равенство $T_{\max i} = T_{\min i}$

Критическим путем графа является множество последовательных узлов, начинающихся входной вершиной и заканчивающихся выходной вершиной, удовлетворяющих условию $T_{\max i} = T_{\min i}$, причем для каждой пары узлов принадлежащих КП соотношения (1) и (2) определяются соседней вершиной в паре.

Длина критического пути определяет минимальное возможное время выполнения графа задачи на МВС.

Граф задачи при выполнении ее на МВС с различной организацией может иметь различные критические пути вследствие изменения времени передач между узлами. Последнее определяется способом организации памяти. Например, в МВС только с общей памятью (без локальной памяти процессоров) при определении критического пути время передачи данных от j -го узла i -му и от i -го узла j -му необходимо удваивать. Двойной учет этого времени происходит потому, что при передаче данных, во-первых, необходимо время τ_{ji} , чтобы передать данные j -го узла в общую память, и, во-вторых, время τ_{ji} чтобы забрать данные из общей памяти и передать i -му узлу, даже в случае выполнения i -го и j -го узлов в одном процессоре. В МВС с распределенной памятью каждый процессор имеет локальную память, в которой могут храниться промежуточные результаты. При этом если узлы j и i обрабатываются одним процессором, то время обмена между ними считается равным 0 (то есть $T_{ij} = 0$), если разными, то время обмена между узлами равно τ_{ji} . Выполнение этих дополнительных условий должно учитываться при решении задачи назначения.

Рассмотрим пример поиска критического пути для графа задачи без учета времен передач (рис. 2). Цифра внутри каждого кружка (узел графа) - номер узла, цифра над кружком - время выполнения узла t_j . Дуги графа не взвешены, следовательно, $T_{ji} = T_{ir} = 0$.

Основным допущением при поиске КП на графе является следующее; граф задачи реализуется на системе с неограниченными ресурсами, в данном случае на МВС с неограниченным числом процессоров.

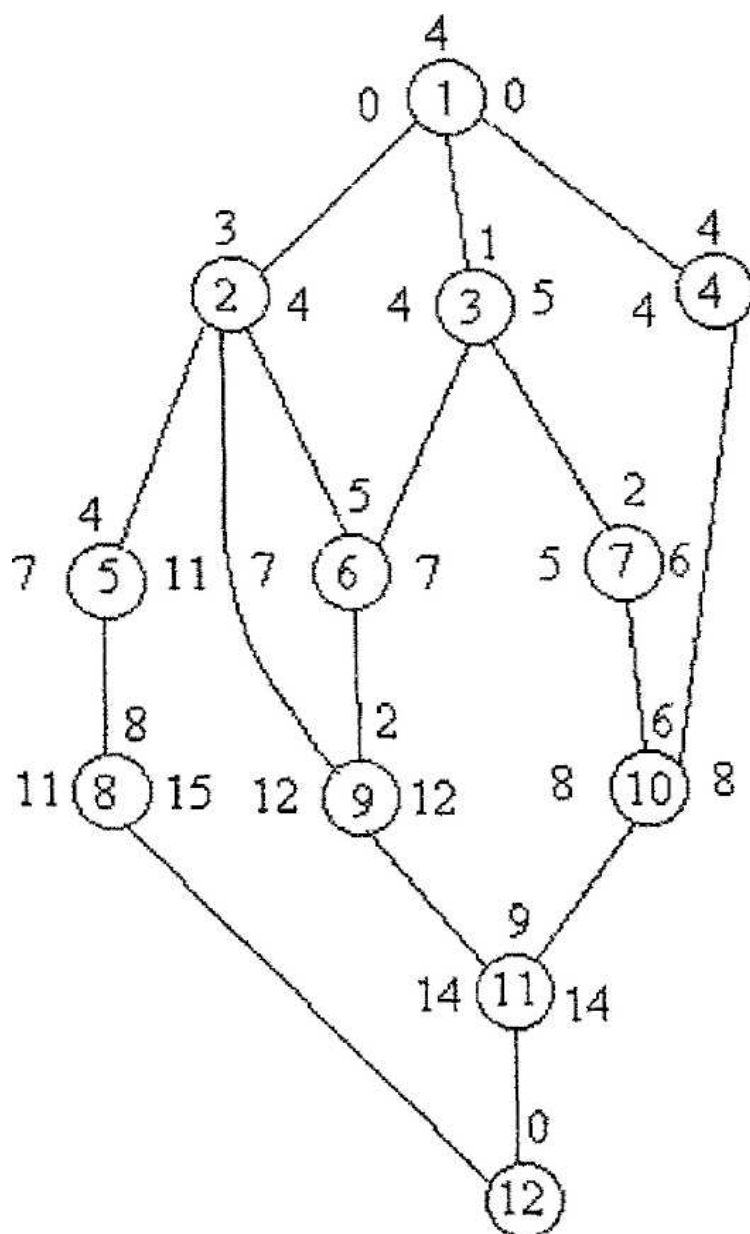


Рис. 2. Определение критического пути для графа задачи без учета передач

При движении по графу от начальной вершины к конечной слева от узла записывается минимально возможное время начала выполнения этого узла $T_{\min i}$. Например, узел 6-может начать выполняться, когда выполнятся 2 и 3 узлы. Тогда $T_{\min 6} = 7$, так как минимально возможное время начала выполнения 6-го узла является максимальным из времен окончания 2-го ($T_{\min 2} + t_2$) и 3-го ($T_{\min 3} + t_3$) узлов.

При движении по графу назад от конечной вершины к первой, справа от узла записывается максимально возможное время начала выполнения этого узла $T_{\max i}$. Например, для узла 2 по формуле (2) находим

$$T_{\max 5} - t_2 = 11 - 3 = 8;$$

$$T_{\max 6} - t_2 = 7 - 3 = 4.$$

$$\text{Следовательно, } T_{\max 2} = \min(T_{\max 5}, T_{\max 6}) = 4.$$

Таким образом, для графа, изображенного на рис. 2, имеются два критических пути 1; 2; 6; 9; 11; 12 и 1; 4; 10; 11; 12. При этом путь 1; 2; 9; 11; 12 не является критическим, в частности, потому что для пары узлов 2 и 9 соотношения (1) и (2) не определяют соседним в паре узлом. Например, $T_{\min 9}$ определяется не узлом 2, а узлом 6, аналогично и $T_{\max 2}$ не определяется узлом 9, т.е.

$$T_{\min 9} \neq T_{\min 2} + t_2 = 4 + 3 = 7;$$

$$T_{\max 2} \neq T_{\max 9} - t_2 = 12 - 3 = 9.$$

$$\text{Следовательно, для данной прикладной задачи } T_{\min} = 23, T_{\max} = 48.$$

Рассмотрим теперь **пример определения КП для графа среднесвязанной задачи с учетом передач** (рис. 3). Дуги графа задачи взвешены числами, соответствующими временам занятости шины.

Поиск критического пути на графе, а следовательно, и минимального времени его выполнения с учетом времени передач данных от узла к узлу, в общем случае, сводится к задаче полного перебора. Тем не менее, вводя ряд ограничений на условие передачи данных, а они связаны с организацией обменов между процессорами и памятью, можно предложить алгоритмы, сокращающие полный перебор и дающие искомый результат.

Сделаем следующие допущения.

- Граф задачи реализуется на системе с неограниченными ресурсами.
- Каждый узел графа может передавать данные по всем выходящим ветвям параллельно и принимать данные по всем входящим ветвям параллельно.

При этих условиях T_{\min} определяется, как было указано выше, в зависимости от способа организации памяти, поскольку этим, в частности, определяются времена передачи между узлами.

- Для случая МВС с общей памятью времена передачи данных удваиваются.

Рассматривая рис. 3, можно сделать вывод, что критический путь графа с учетом передач изменился - 1; 3; 7; 10; 11; 12, а $T_{\min} = 46$.

Рассмотрим теперь алгоритм поиска КП для графа задачи с учетом передач на МВС с распределенной памятью (рис. 4). Отметим, что рассматриваемая задача аналогична задаче, граф которой приведен на рис.3. Следует обратить внимание на то, что целью вычисления КП при реализации задачи на МВС с распределенной памятью является определение среди узлов-последователей ($r=1...v$) (см. рис. 1) узла, который назначается на процессор, выполняющий узел-предшественник. В этом случае данные от узла-предшественника не передаются узлу-последователю (T_{jr} приравнивается к нулю).

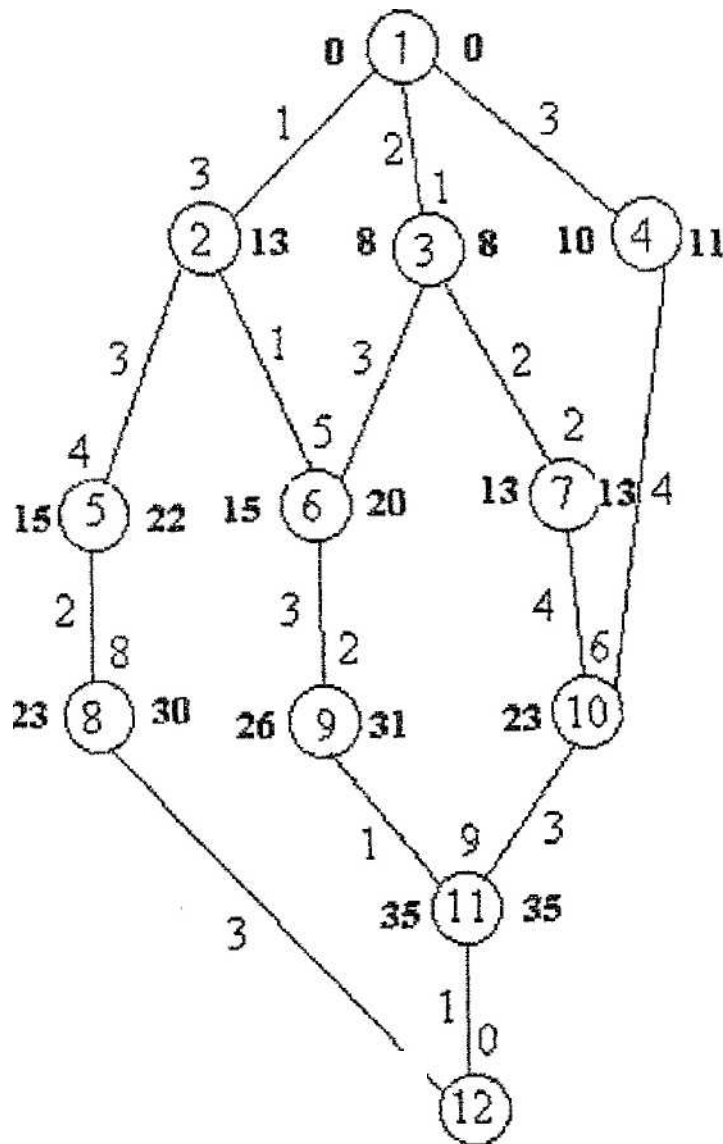


Рис. 3. Определение критического пути для графа задачи с учетом передач на МВС с общей памятью

Следующие допущения являются основными при реализации графа задачи на МВС с распределенной памятью.

1. Граф реализуется на системе с неограниченными ресурсами.
2. От каждого узла графа данные могут передаваться параллельно по исходящим ветвям.

3. Каждый узел графа может принимать данные параллельно по входящим ветвям.
4. Каждый узел графа может иметь не больше одной входящей и не больше одной исходящей ветви, времени передачи по которой в результате поиска КП присваивается значение равное нулю.

Алгоритм определения КП для графа задачи, выполняемой на МВС с распределенной памятью, состоит в том, что при вычислении по формуле (1) минимально возможного времени начала выполнения i -го узла $T_{\min i}$ определяется одна из входящих в него ветвей, времени передачи по которой присваивается значение равное нулю. При этом это должна быть такая ветвь, которая вносит максимальный эффект в уменьшении $T_{\min i}$. Очевидно, что не всегда с первой же попытки можно определить $T_{\min i}$. Например, определив ветвь d_i , входящую в i -й узел, обуславливающую $T_{\min i}$ необходимо провести анализ узла d , из которого исходит эта ветвь. Если у этого узла уже есть одна исходящая ветвь, которой

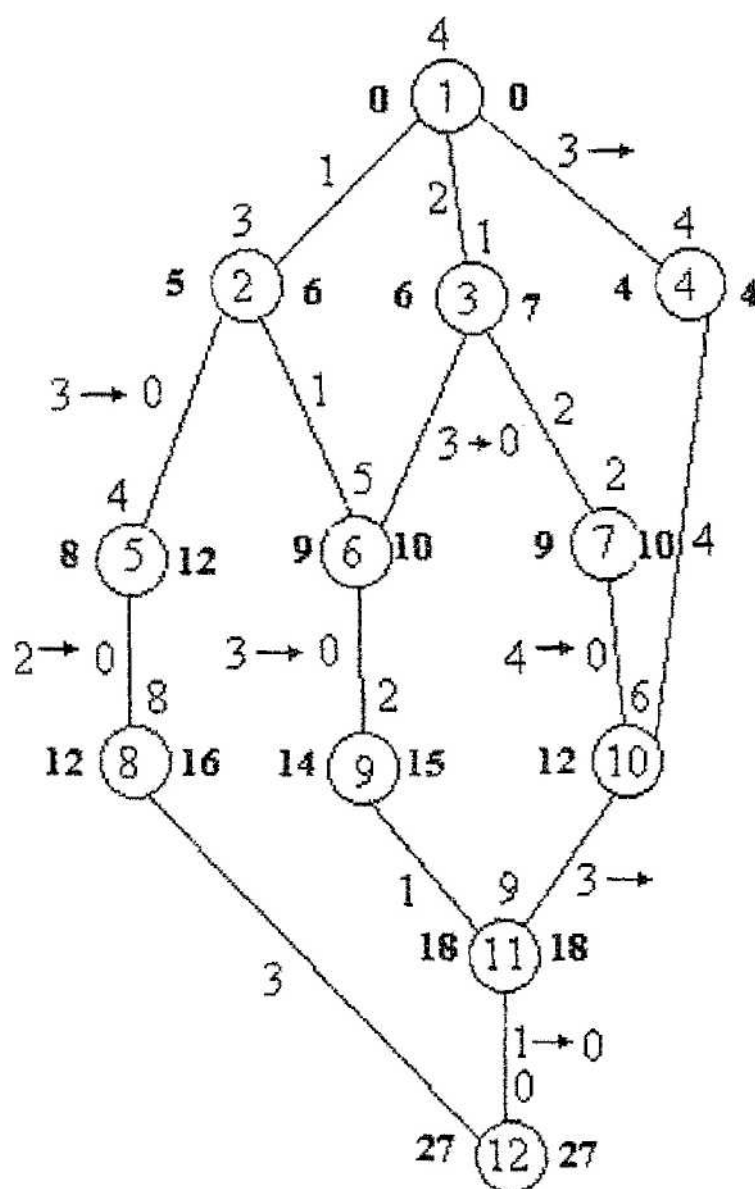


Рис. 4. Определение критического пути для графа задачи с учетом передач на МВС с распределенной памятью

присвоено значение $T_{dk}=0$ (т.е. определен k -й узел - последователь d -го узла, и, следовательно, процессор оставил необходимые данные для выполнения k -го узла в своей локальной памяти), то ветвь di из рассмотрения исключается. Находится следующая по значению эффективности ветвь, входящая в i -й узел и т.д.

Если у d -го узла нет исходящей ветви, которой присвоено значение равное нулю, то для этого d -го узла определяется присутствие такой ветви, у которой $\tau_{dk} > \tau_{di}$. Если такая ветвь есть, то i -й узел исключается из рассмотрения до тех пор, пока не будет принято решение о присвоении или не присвоении ветви dk значения $T_{dk}=0$. Если такой ветви нет, то T_{di} присваивается значение, равное нулю, и уже с учетом этого определяется $T_{\min i}$.

Следует обратить внимание на следующее. Если k -й узел расположен на более низком ярусе графа чем i -й узел, то при нахождении $T_{\min k}$ анализируются все пути от вершины d к вершине k , а значение 0 может быть присвоено (или не присвоено) в соответствии с правилами, изложенными выше для узла i , одной из ветвей, исходящих из узла d (необязательно для ветви dk).

Максимально возможное время начала выполнения узла $T_{\max i}$ вычисляется по формуле (2) при обратном движении по графу уже с учетом определенных выше нулевых ветвей. Данные ветви помечены на рис. 4 стрелкой (например, $3 \rightarrow 0$). В приведенном примере граф задачи имеет единственный КП 1-4-10-11-12, длина которого равна $T_{\min}=27$.

Из рассмотренных примеров видно, что длина критического пути (T_{\min}) является основным параметром, характеризующим информационно-логическую структуру вычислительного процесса (ВП), реализующего решение заданной прикладной задачи. В случае учета времени передачи данных от одного узла к другому при решении задачи в МВС с ограниченными ресурсами, значение минимального времени ее выполнения увеличивается и определяется характеристиками реальной структуры МВС. Очевидно, что это увеличение будет тем меньше, чем удачнее распределены узлы графа по процессорам МВС.

Таким образом, возникает задача построения оптимального расписания (назначения готовых к исполнению узлов ВП на свободные процессоры).

3. Постановка задачи назначения

Прежде чем рассматривать существо задачи назначения, отметим, что задача назначения может решаться как в **статическом**, так и в **динамическом** режимах.

В первом случае задача назначения выполняется до начала реализации ВП в МВС (именно этот случай исследуется в данном цикле лабораторных работ), во втором она выполняется непосредственно в процессе реализации ВП.

Теоретической основой решения задачи назначения является теория расписаний.

Под решением задачи назначения понимается процесс распределения узлов графа задачи (набора задач), выполняемой в МВС, между ее процессорами, при котором определяется время начала выполнения узла, его длительность и назначение процессора, который обеспечит это выполнение.

Модель процесса распределения включает средства, описывающие ресурсы, систему узлов и дуг графа задачи (набора задач) и критерий оптимальности распределения. Под ресурсами понимаются: модули обработки (процессоры), модули памяти (она может быть распределенной, общей или смешанной), внутрисистемный интерфейс (общая шина, мультиплексная шина). При построении алгоритмов назначения в отказоустойчивых МВС в модель должны быть введены средства, описывающие систему обеспечения отказоустойчивости МВС. Например, средства, обеспечивающие введение дополнительных копий узлов графа задачи и дополнительных процессоров.

Рассмотрим наиболее простой случай.

Пусть в качестве ресурсов в модели используется только набор однотипных процессоров, имеющих равное быстродействие. На данном наборе процессоров выполняется вычислительный процесс, имеющий сетевую структуру и представляющий собой совокупность отдельных алгоритмов (сегментов задачи) $Z = \{z_i\}$.

Формально модель выполнения задачи Z можно представить совокупностью

$$\{Z, <, T, W, \Theta\},$$

где $Z = \{z_i, z_L\}$ - множество сегментов задачи, выполняемых в системе (узлы графа);

$<$ - означает задание в множестве Z отношения частичного порядка, которое определяет последовательность выполнения сегментов и информационные связи между ними (связность узлов);

$T = \{t_1, \dots, t_L\}$ - вектор времен выполнения сегментов на процессоре с заданным быстродействием;

$W = \{w_1, \dots, w_L\}$ - вектор коэффициентов важности сегментов, соответствующих коэффициентам относительных потерь эффективности из-за невыполнения сегментов задачи вследствие отказа процессора, на котором выполняется данный сегмент;

$\|\Theta\| = \|\tau_{iq}\|, i = 1 \dots L, q = 1 \dots L-1$ - матрица времен занятости шины с заданной пропускной способностью при передаче данных между узлами i и q .

Величина τ_{iq} является характеристикой не только структуры графа задачи и пропускной способности шины, но и способа организации МВС.

Как было отмечено выше, в качестве модели задачи используется направленный граф, узлы которого отображают сегменты задачи. При построении алгоритмов, реализующих задачу назначения, направленный граф преобразуется в таблицу связности, элементы которой

$$A_{ij} = \begin{cases} 1, & \text{если выходная информация узла } z_i \text{ является входной} \\ & \text{для узла } z_j \\ 0 & \text{в противном случае.} \end{cases}$$

В качестве критериев оптимальности распределения узлов в МВС применяются обычно либо минимум времени выполнения задачи (набора задач) при ограничении на число процессоров, либо минимум числа требуемых процессоров при ограничении на время решения задачи (набора задач).

В качестве производных от этих критериев используются следующие –

- максимум загрузки каждого процессора ($K_{\text{загр } i}$),
- минимум простоев каждого процессора ($K_{\text{пр } i}$).

$$K_{\text{загр } i} = T_i / T_{\text{вып}}, \quad (3)$$

где T_i – время, в течение которого i -й процессор занят обработкой задачи;
 $T_{\text{вып}}$ - время выполнения задачи;

$$K_{\text{пр } i} = 1 - K_{\text{загр } i}. \quad (4)$$

Следует отметить при этом, что эффективность как алгоритмов распределения узлов задачи, так и выбранной структуры МВС можно оценить с помощью коэффициента ускорения ($K_{\text{уск}}$), показывающего ускорение времени решения задачи на n процессорах в сравнении с временем решения этой же задачи на одном процессоре

$$K_{\text{уск}} = T_{\text{max}} / T_n, \quad (5)$$

где T_{max} – время решения задачи на одном процессоре;
 T_n – время решения задачи на n процессорах.

В теоретическом плане при построении алгоритмов оптимальных расписаний могут быть использованы математические

методы, например, динамического программирования. Однако это достаточно сложная проблема, так как необходимо решать задачи большой размерности, при этом они относятся к NP -полным. На практике оптимальные расписания с использованием таких методов построены для простых типов прикладных задач сравнительно небольшой размерности, причем в основном для двухпроцессорных систем. Поэтому обычно применяются методы приближенной оптимизации, в частности, эвристические методы. Эвристические методы построения расписаний характеризуются тем, что они приспособляются как к информационно-логической структуре ВП, так и к структурной организации МВС, и относятся к классу приоритетного распределения. В таких расписаниях узлам задачи присваивается приоритет по тем или иным правилам (стратегии назначения), после чего узлы упорядочиваются в виде линейного списка по убыванию приоритетов. В процессе составления расписания осуществляется назначение узлов на процессоры в соответствии с их приоритетами для их выполнения. Наиболее исследованы и представлены в литературе различными моделями следующие стратегии назначения готовых к выполнению узлов вычислительного процесса:

- 1) равновероятный выбор;
- 2) выбор узла с минимальным временем выполнения;
- 3) выбор узла с максимальным временем выполнения;
- 4) выбор узла, принадлежащего критическому пути;
- 5) выбор узла, имеющего наибольшее число связей с последующими узлами;
- 6) выбор узла в порядке поступления в очередь на исполнение.

В данном цикле лабораторных работ используются стратегии 2 - 5 , при этом в ряде работ, где исследуются вопросы обработки набора задач, выбор готового к исполнению узла осуществляется с учетом приоритета задачи.

При выполнении лабораторных работ постановка задачи организации параллельных вычислений в МВС сводится к реализации следующей целевой функции:

- ◆ определить минимальное число процессоров,
- ◆ шин связи коммутационной сети,
- ◆ модулей памяти,
- ◆ способ организации памяти и
- ◆ тип стратегии назначения, обеспечивающих выполнение прикладной задачи (набора задач) за заданное время.

Таким образом, при реализации выбранной стратегии назначения узлы ВП $Z-\{z_i\}$, $i=1...L$ статически распределяются по процессорам МВС

так, что каждому из выбранного числа процессоров M_j сопоставляется некоторое подмножество узлов $z_i \in Z$. Результатом распределения сегментов задач по процессорам является матрица $\|X\| = \|X_{ij}\|$, ($i=1...L$, $j=1...n$) и временная диаграмма занятости процессоров и шин, число которых тоже выбрано при выполнении поставленной задачи.