

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Цикл лабораторных работ посвящен одному из основных разделов курса “Вычислительные системы”, связанному с организацией параллельных вычислений в многопроцессорных вычислительных системах (МВС). Рассматриваемые системы относятся к классу “множественный поток команд, множественный поток данных” (МКМД) [1,2]. 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 процессоров 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) – тип (общая шина, мультиплексная шина), число шин, пропускная способность, характеризующая относительным временем занятия сформированного канала (процессор – процессор, процессор – общая память, общая память – процессор) при передаче данных;

- 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

части (ветви) программы, начав выполнение которых процессор заканчивает их без прерываний, а дуги – связи по данным между 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_{Π}). В соответствие с этим соотношением различают слабосвязанные задачи ($t_p \gg t_{\Pi}$), среднесвязанные ($t_p \approx t_{\Pi}$), сильносвязанные ($t_p \ll t_{\Pi}$).

Каждая задача может характеризоваться максимальным временем выполнения T_{\max} и минимальным временем T_{\min} . Например, для задач, у которых для всех узлов $t_{\Pi} = 0$, 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, применяемых при выполнении данного цикла лабораторных работ.

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 [3] 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

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

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.

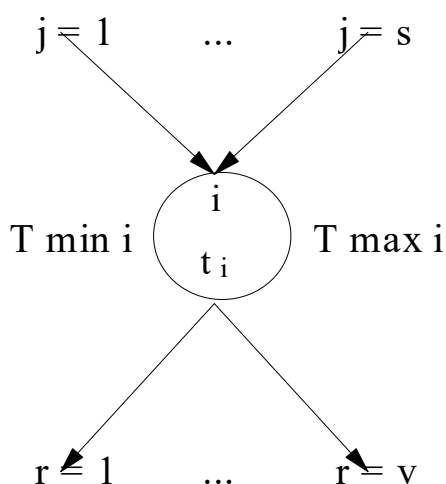


Рис. 1. Пояснения к определению критических узлов

При повторном анализе графа ВП 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_{ji} .

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

$$T_{\min i} = T_{\max i}.$$

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

При этом необходимо отметить следующее. Длиной критического пути определяется минимальное время выполнения задачи. 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 r -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 - $T_{ji}=0$, если разными, то время обмена между узлами равно τ_{ji} . Выполнение этих дополнительных условий должно учитываться при решении задачи назначения (см. ниже).

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_i . 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.

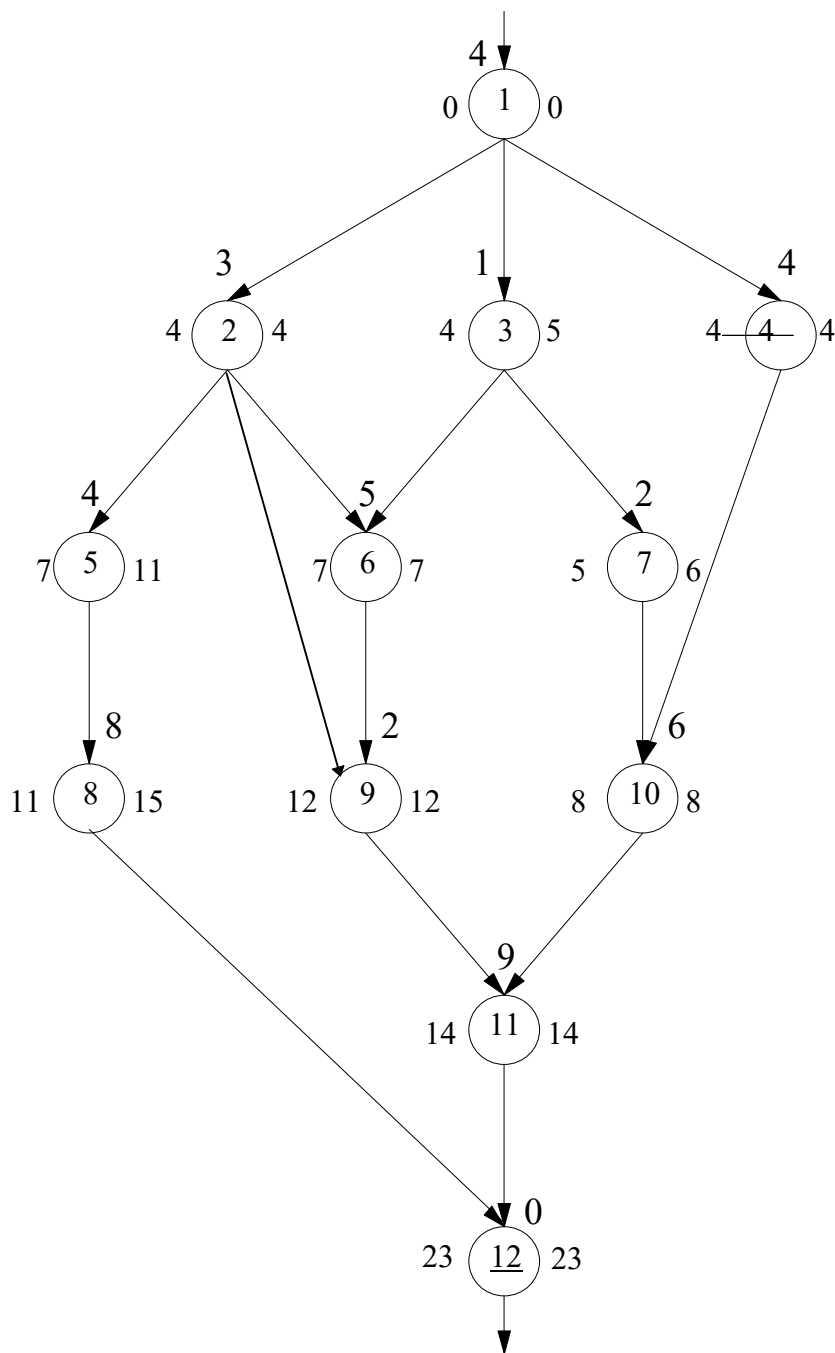


Рис. 2. Определение критического пути для графа задачи без учета передач

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.

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

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

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} определяет-

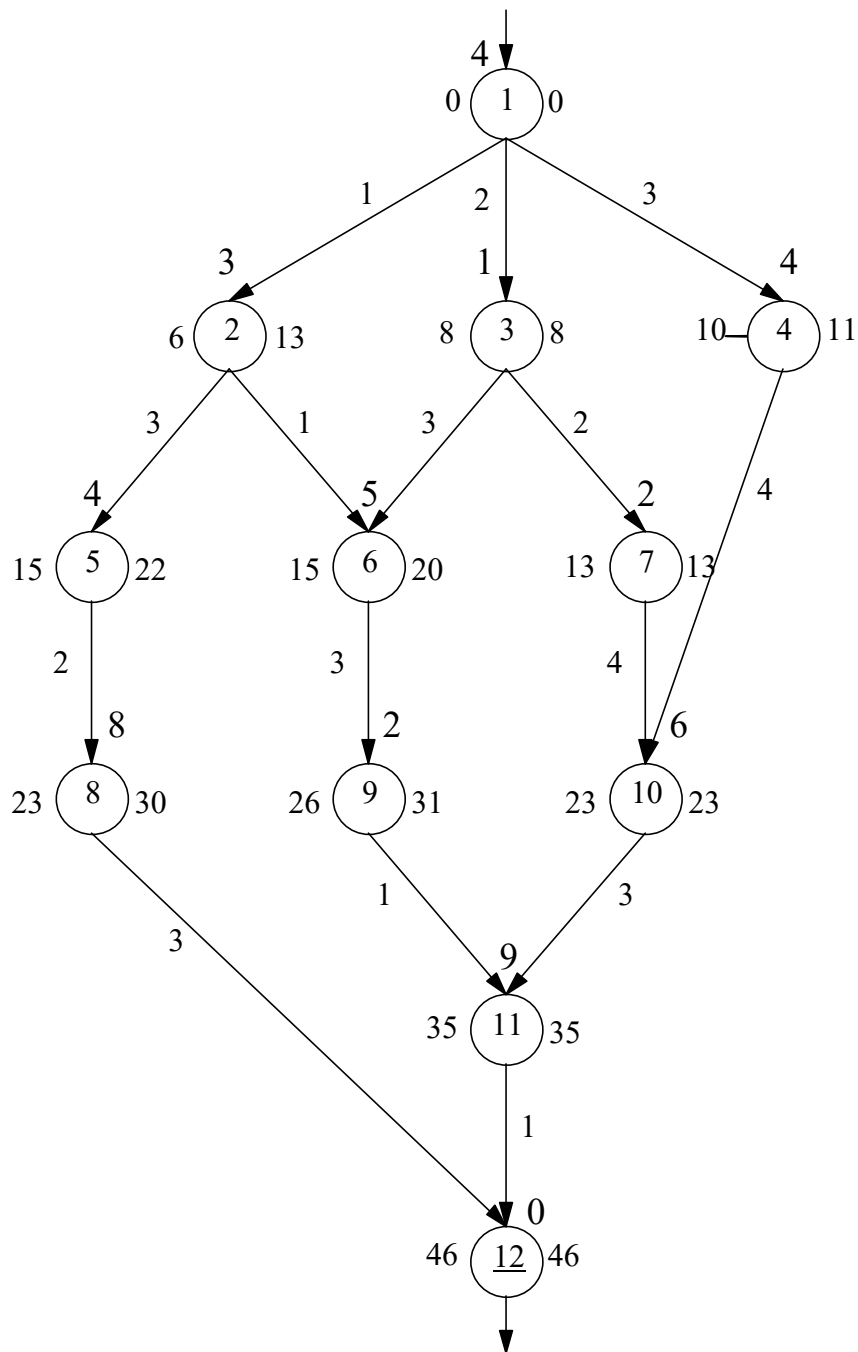


Рис. 3. Определение критического пути для графа задачи с учетом передач на МВС с общей памятью

ся, 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 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 ($r=1 \dots v$) (см. рис. 1) узла, который назначается на процессор, выполняющий узел-предшественник. В этом случае данные от узла-предшественника не передаются узлу-последователю (T_{ir} приравнивается к нулю).

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

1. Граф задачи 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 критического пути 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 ветвь, 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, ко-

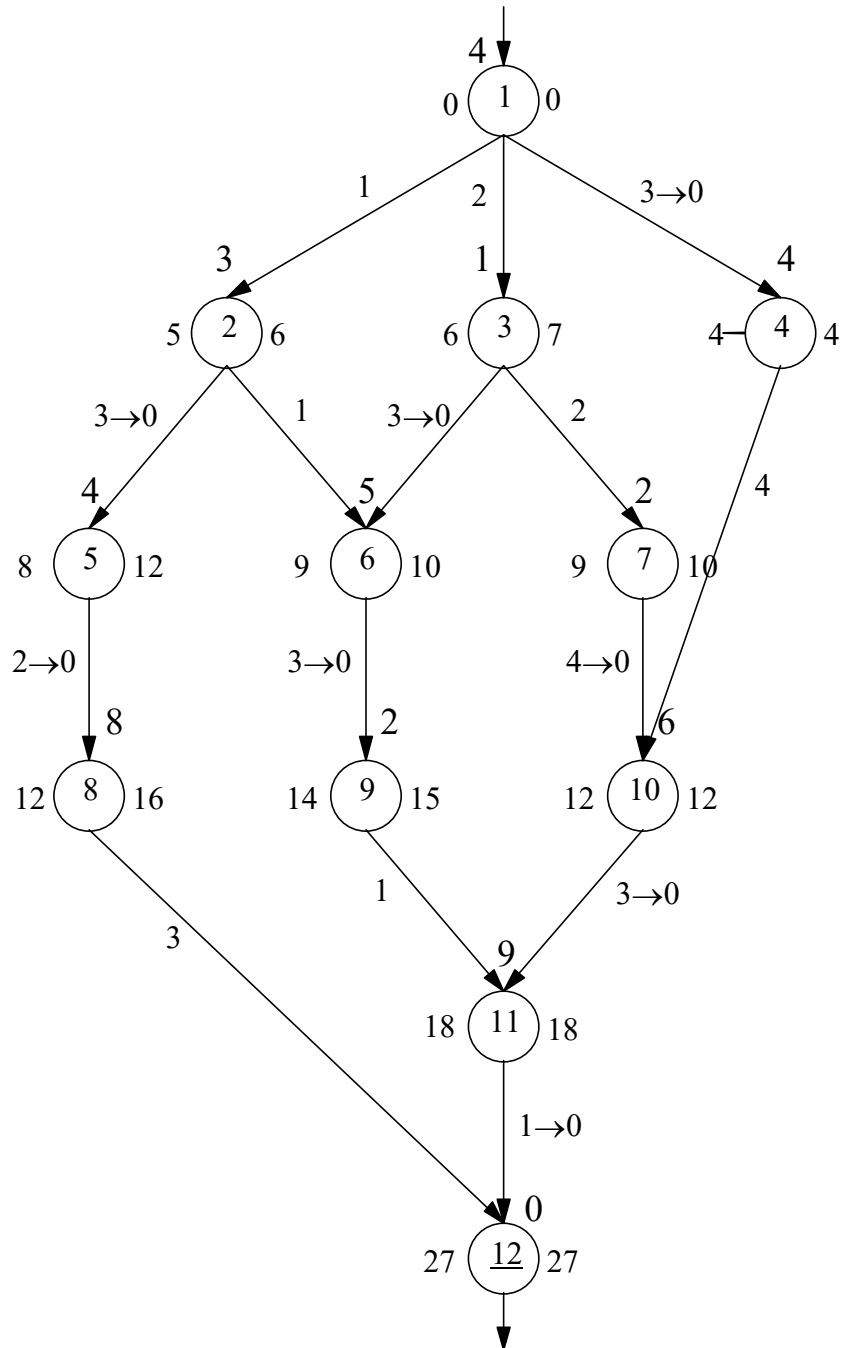


Рис. 4. Определение критического пути для графа задачи с учетом передач на МВС с распределенной памятью

торой 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 [2] 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 его 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).

Постановка задачи назначения

Прежде чем рассматривать существо задачи назначения, отметим, что она 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 [3,4]. 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 используются 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 $Z = \{ z_1, \dots, z_L \}$ – 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;

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

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

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 [3,4]. 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 стра-

тегии назначения, обеспечивающих выполнение прикладной задачи (набора задач) за заданное время.

Таким образом, при реализации выбранной стратегии назначения узлы ВП $Z = \{z_i\}$, $i=1 \dots L$ статически распределяются по процессорам МВС так, что каждому из выбранного числа процессоров M_j сопоставляется некоторое подмножество узлов $z_j \in Z$. Результатом распределения сегментов задач по процессорам является матрица $\|X\| = \|x_{ij}\|$, $i = 1 \dots L$, $j = 1 \dots n$ и временная диаграмма занятости процессоров и шин, число которых тоже выбрано при выполнении поставленной задачи.

Особенности выполнения лабораторных работ

Целью исследований в первой лабораторной работе является изучение принципов распределения узлов ВП в однозадачном режиме, анализ временных диаграмм с целью определения времен простоя процессоров из-за неготовности узлов задач. Интервалы простоя назовем “пустотами”. В этих “пустотах” могут быть выполнены копии сегментов, готовых к выполнению. Тем самым повышается вероятность обнаружения ошибок в МВС, которая рассчитывается по формуле

$$P_{\text{ош}} = \Sigma T_{\text{дубл}} / T_{\text{max}}, \quad (6)$$

где $\Sigma T_{\text{дубл}}$ – суммарное время выполнения копий узлов задач;

T_{max} – суммарное время выполнения всех узлов.

В процессе выполнения лабораторной работы № 1 необходимо построить временные диаграммы выполнения вычислительного процесса, проанализировать “пустоты” и определить оптимальное множество копий узлов для достижения максимальной вероятности обнаружения ошибки при заданном времени выполнения ВП.

Рассмотрим пример выполнения некоторых пунктов задания лабораторной работы № 1 относительно графа задачи, приведенного на рис. 2.

Выше было получено, что для этой задачи $T_{\text{max}}=48$, $T_{\text{min}}=23$, следовательно, первое приближение числа процессоров МВС, с помощью которых можно выполнить задачу за минимальное время $n=3$ (ближайшее большее целое от значения $T_{\text{max}} / T_{\text{min}}$). Примем в качестве стратегии назначения – выбор готового к исполнению узла ВП с максимальным временем выполнения. Временная диаграмма выполнения задачи, соответствующая данной

стратегии, представлена на рис. 5. (В случае нескольких свободных процессоров назначение готового к исполнению узла осуществляется на процессор с меньшим номером). По вертикальной оси - 1,2,3 - номера процессоров, по горизонтальной - время, заданное в условных единицах, в интервалах занятости процессоров проставлены номера узлов задачи. Анализируя временную диаграмму, можно сделать следующие выводы.

1. Минимальное число процессоров, с помощью которых задача выполняется за $T_{\min} - n_{\min} = 3$, так как один из процессоров (первый) работает без простоев, два других заняты без перерывов.

2. Коэффициенты загрузки процессоров, вычисленные по формуле (3) соответственно равны

$$K_{z1}=1, K_{z2}=10/23, K_{z3}=15/23.$$

3. Коэффициент ускорения, определенный по формуле (5) (максимально возможный для данной задачи) $K_y = 48/23$.

4. Заштрихованные интервалы времени (“пустоты”) для второго и третьего процессоров можно использовать для выполнения копий узлов ВП, готовых к исполнению на начало интервала простоя процессора. Следует обратить внимание на то, что при заполнении “пустот”, с точки зрения контроля всех процессоров, целесообразно, чтобы копии одинаковых узлов выполнялись на разных процессорах. Соответствующая диаграмма представлена на рис. 6.

На основании результатов ее анализа определим вероятность определения ошибки $P_{\text{ош}}$

$$P_{\text{ош}} = (4+9+4) / 48 = 17/48.$$

Во второй лабораторной работе в отличие от первой изучаются принципы распределения узлов ВП в многозадачном режиме. Так же как и в однозадачном режиме готовый к исполнению узел ВП, выбранный с учетом приоритета задачи и стратегии назначения, начинает выполняться на свободном процессоре без временных задержек независимо от того, какой задаче он принадлежит. В многозадачном режиме максимальное время выполнения набора задач равно сумме времен выполнения всех узлов всех задач набора. Минимальное время выполнения набора задач равно максимальному значению из минимальных времен выполнения каждой задачи. Следует обратить внимание на то, что в многозадачном

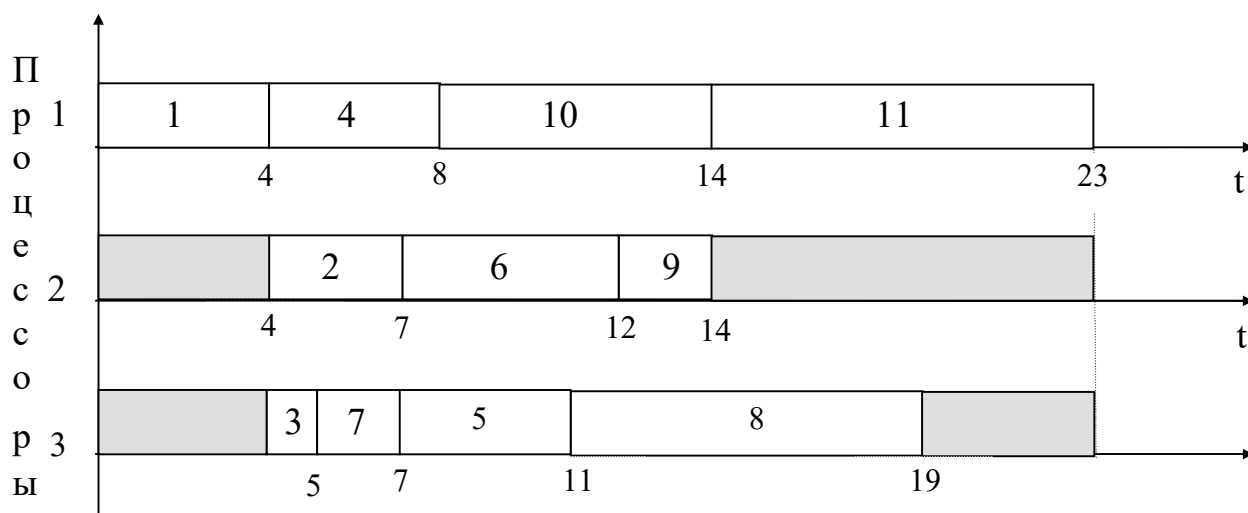


Рис. 5. Диаграмма выполнения задачи без учета передач

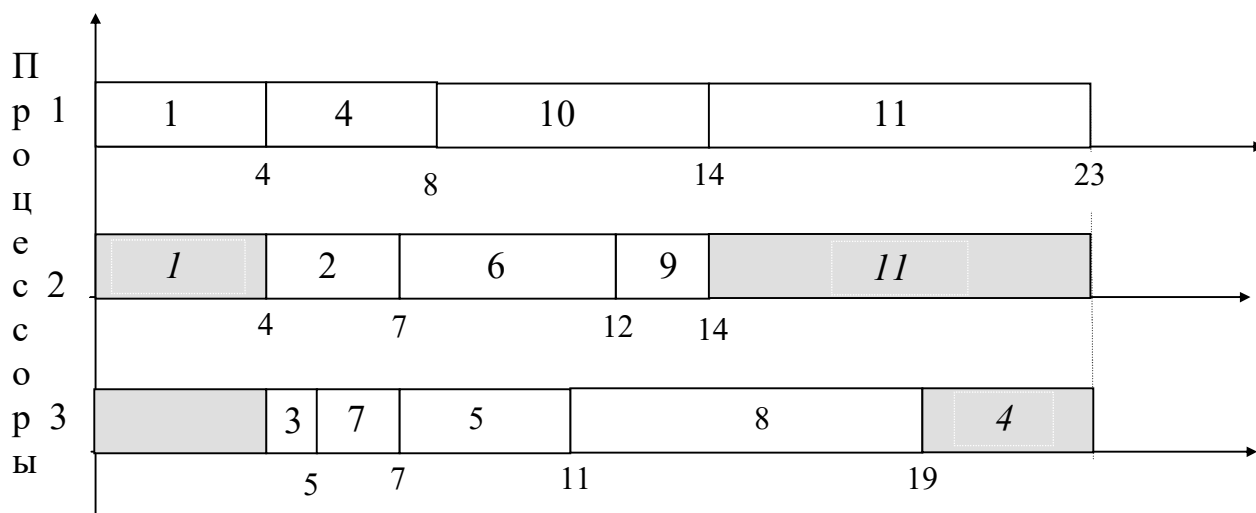


Рис. 6. Пример выполнения копий узлов в "пустотах" временной диаграммы

режиме с учетом приоритетов среди готовых к исполнению узлов ВП формируются отдельные очереди на выполнение в соответствии с приоритетами. Первыми назначаются на выполнение узлы из очереди с наивысшим приоритетом, в соответствии с выбранной стратегией. Если эта очередь пуста, то анализируется очередь с меньшим приоритетом и т.д.

Целью исследований в лабораторной работе №3 является изучение организации параллельных вычислений в МВС с общей памятью (рис. 7) в многозадачном режиме.

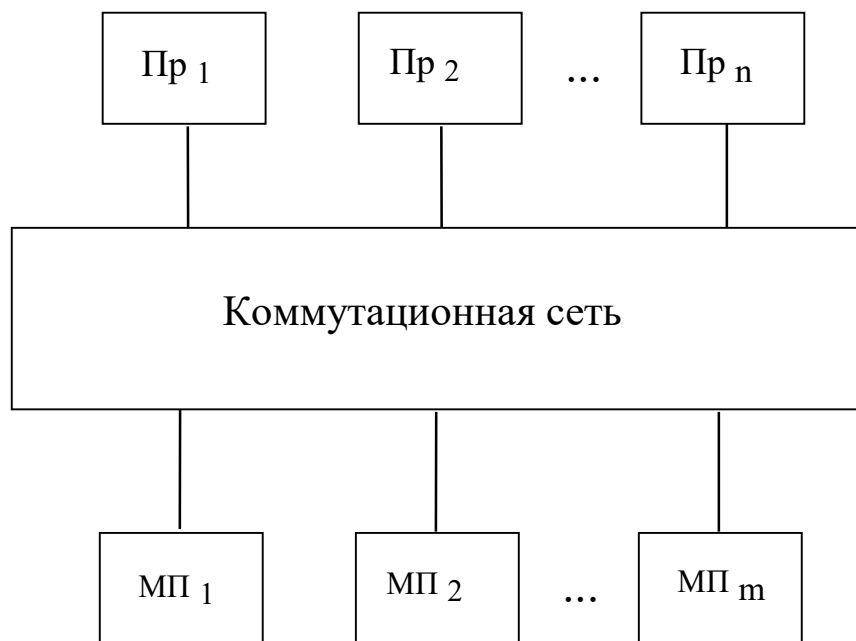


Рис. 7. МВС с общей памятью

Следует обратить внимание на следующее.

1. МВС состоит из n процессоров, m шин, связывающих каждый процессор с каждым из модулей памяти.

2. Процессор, выполнив очередной узел ВП, может “захватить” любую свободную шину и передать все результаты обработки в любой один свободный с точки зрения процесса записи/считывания модуль памяти (емкость модуля памяти считается неограниченной).

3. Процессор, на который назначен для исполнения узел ВП, может “захватить” любую свободную шину, и если модуль памяти, в котором находятся исходные данные для этого узла не занят записью/считыванием данных другим процессором, то происходит их считывание.

Рис.8. Диаграмма выполнения задачи в МВС с общей памятью
4. При реализации стратегии назначения учитываются только времена выполнения узлов задачи.

Рассмотрим пример построения временной диаграммы (рис. 8) выполнения только одной задачи, граф которой представлен на рис. 3, в МВС, у которой $n=3$, $m=2$.

На временной диаграмме рис. 8 прямоугольники, отмеченные одной цифрой, - это интервалы времени обработки соответствующего узла задачи, двумя цифрами - интервалы времени, соответствующие передаче данных от одного узла задачи (верхняя цифра) к другому (нижняя цифра) по тракту процессор-шина-память. Время выполнения задачи на МВС с общей памятью равно 61 единицам времени.

Лабораторная работа №4 посвящена изучению организации параллельных вычислений в МВС с распределенной памятью (рис.9).

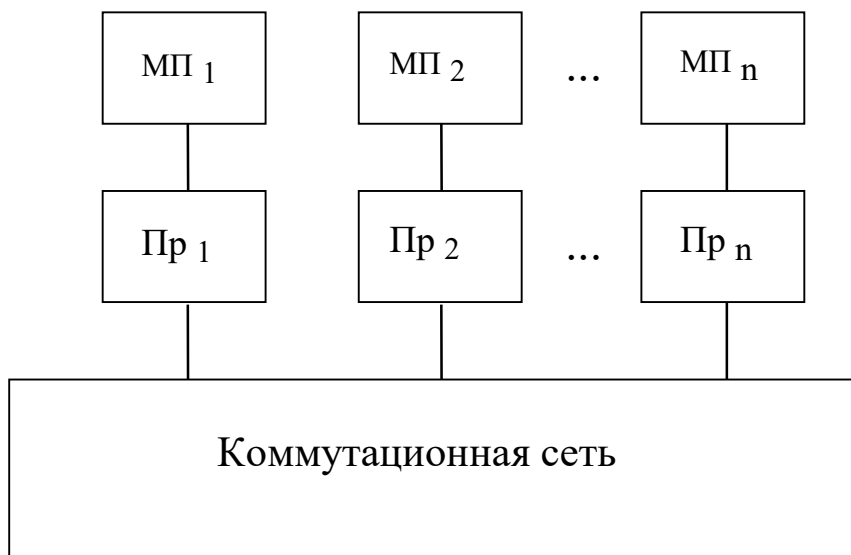


Рис. 9. МВС с распределенной памятью

Рассмотрим МВС изображенную на рис. 9 со следующими характеристиками. Каждый из n процессоров имеет свою локальную память, емкостью достаточной для хранения промежуточных результатов обработки любого узла задачи. Каждый процессор имеет доступ к памяти любого другого процессора по считыванию из нее данных в свою память по одной из m шин, при этом $m < n/2$. Одновременно запись в память или считывания из нее осуществляется только по одному каналу.

Рис. 10. Диаграмма выполнения задачи в МВС с распределенной памятью

На рис. 10 представлена временная диаграмма выполнения задачи, представленной на рис. 4, в соответствие со стратегией назначения готового

к исполнению узла с максимальным временем выполнения. Однако в данном примере на процессор, выполнивший очередной узел, может быть назначен узел, следующий за ним и не нуждающийся в приеме данных от других узлов. Так второй процессор за 5-м узлом выполняет 8-й, хотя 1-й процессор закончил выполнение 4-го узла ранее и в тот момент готовым к исполнению был только узел 7. Поэтому он был назначен на 1-й процессор.

Перед началом каждой лабораторной работы необходимо получить у преподавателя исходные данные для ее выполнения. При выполнении лабораторных работ используются программные модели, написанные на языке Turbo-Pascal. При запуске программ на экране появляется стандартное меню, включающее блок ввода исходных данных, блок моделирования и блок просмотра результатов. При вводе исходных данных необходимо последовательно ввести следующие данные: режим работы МВС (однозадачный или многозадачный); граф задачи (задач), который вводится из файла данных; характеристики системы: количество процессоров, шин, модулей памяти; стратегию выбора готовых узлов. В блоке просмотра результатов приводятся временные характеристики ВП, распределение узлов графа по процессорам, временные диаграммы выполнения ВП, коэффициенты загрузки процессоров, шин и т.д.

Полученные результаты моделирования используются для исследований при выполнении лабораторных работ.

Лабораторная работа № 1

ИССЛЕДОВАНИЕ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЙ В МНОГОПРОЦЕССОРНОЙ ВС

Цель работы : изучение принципов распределения узлов вычислительного процесса (ВП) в однозадачном режиме с использованием различных стратегий назначения готовых к исполнению узлов ВП.

Домашняя подготовка

1. Изучить соответствующий раздел лекционного курса и настоящее описание лабораторных работ. По номеру бригады получить у преподавателя исходные данные для исследования - граф задачи 1, ограничения на время ее выполнения.

2. На заданном графе вычислительного процесса с использованием алгоритма, приведенного выше, рассчитать следующие характеристики:

2.1. Критические пути - методом определения минимального и максимального времени начала выполнения узлов ВП путем движения вниз и обратного движения по графу ВП.

2.2. Временные характеристики ВП - максимальное и минимальное время выполнения графа ВП.

3. С использованием различных стратегий распределить узлы ВП по процессорам для выполнения ВП за минимальное время, определив необходимое для этого количество процессоров, коэффициенты их загрузки.

Построить временную диаграмму.

Лабораторное задание

1. Промоделировать исследуемый граф с использованием различных стратегий выбора готовых узлов ВП для достижения заданного времени выполнения графа ВП.

Результаты моделирования для различных стратегий выбора готовых узлов свести в таблицу (количество процессоров, время решения задачи, коэффициенты загрузки).

По результатам моделирования построить зависимости $t(n)$ для различных стратегий выбора готовых узлов ВП, где t – время выполнения вычислительного процесса, n – количество процессоров, на которых выполняется ВП.

Сравнить полученные графики, выбрать лучшую стратегию и объяснить результаты.

2. Распределить узлы ВП в режиме максимального заполнения "пустот" временной диаграммы решением копий узлов ВП. Для исследования в п.2 получить у преподавателя граф задачи 2.

Заданное время выполнения задачи 2 определить как

$$T_{\text{зад}} = T_{\text{min}} + 4.$$

2.1. Повторить выполнение п.2.1 домашней подготовки и п.1 лабораторного задания для графа задачи 2.

2.2. При заданном времени выполнения задачи 2 и определенном количестве процессоров достичь максимальной вероятности обнаружения ошибки $P_{\text{ош}}$, выполняя копии узлов в "пустотах" на временной диаграмме.

Проверить правильность выполнения ВП с помощью программного моделирования. Полученные результаты сравнить с временными издержками в случае $P_{\text{ош}}=1$.

Проанализировать и объяснить полученные результаты.

Рекомендации для выполнения п. 2.2.

Для выполнения п.2 перерисовать временные диаграммы выполнения ВП, полученные в п.2.1. для случая выполнения ВП за заданное время на минимальном количестве процессоров.

Выбор копий узлов для заполнения "пустот" производить исходя из условия $P_{\text{ош}} = \Sigma T_{\text{дубл}} / T_{\text{мах}} \rightarrow \text{мах}$,

где $\Sigma T_{\text{дубл}}$ – суммарное время выполнения копий узлов;

$T_{\text{мах}}$ – суммарное время выполнения всех узлов ВП.

Для программного моделирования ВП с учетом копий узлов необходимо произвести изменение исходной матрицы смежности, которая служит формальной моделью графа в используемых программных моделях. Для этого необходимо добавить в матрицу смежности число дополнительных строк и столбцов, равное числу копий узлов ВП, правильно расставить связи между узлами.

Для достижения $P_{\text{ош}}=1$ необходимо продублировать все узлы ВП, т.е. выполнить параллельно две одинаковые задачи в многозадачном режиме.

Отчет должен содержать :

1. Результаты домашней подготовки.
2. Таблицы и графики по п.1 лабораторного задания.
3. Неизбыточный и избыточный графы задачи, временные диаграммы и количественные оценки по п.2 лабораторного задания.

Контрольные вопросы

1. Что понимается под решением задачи назначения?
2. Какие критерии оптимальности распределения узлов ВП применяются при решении задачи назначения?
3. Как определить критический путь для задачи без учета передач? Почему узел является критическим?

4. Что влияет на выбор стратегии назначения готовых к выполнению узлов ВП? Какие стратегии выбора чаще всего используются и почему?

5. Какими преимуществами обладают различные способы повышения вероятности обнаружения ошибок в ВП? Как можно достичь $P_{\text{ош}}=1$?

6. Чему равна максимальная вероятность обнаружения ошибок при выполнении задачи за заданное время на определенном количестве процессоров? Всегда ли можно достичь этого значения?

Лабораторная работа № 2

ОРГАНИЗАЦИЯ МНОГОЗАДАЧНОГО РЕЖИМА ВЫПОЛНЕНИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В МВС

Цель работы: исследование задачи составления расписания выполнения ВП в многозадачном режиме в МВС с использованием различных стратегий назначения готовых узлов, с учетом приоритета выполняемой задачи и без учета приоритета; изучение методики выбора характеристик МВС (количество процессоров, стратегия назначения, приоритетность задачи) для достижения заданного времени выполнения набора задач.

Домашняя подготовка

1. Изучить соответствующий раздел лекционного курса и настоящее описание лабораторных работ. Получить у преподавателя исходные данные для исследования набора из двух задач: графы задач, стратегии выбора готовых узлов, количество процессоров.

2. Для заданного набора задач рассчитать минимальное и максимальное время его выполнения.

Лабораторное задание

1. В однозадачном режиме промоделировать отдельно выполнение каждой задачи с использованием заданной стратегии выбора готовых узлов на заданном количестве процессоров.

Результаты моделирования для каждой задачи свести в таблицу (количество процессоров, время решения задачи, коэффициенты загрузки). Определить суммарное время выполнения двух задач в однозадачном режиме.

2. Выполнить набор из двух задач с равными приоритетами в многозадачном режиме на том же количестве процессоров и с той же стратегией назначения, что и в п.1. Определить время выполнения набора задач и коэффициенты загрузки процессоров.

3. Сравнить результаты, полученные в п.1 и п.2, и вычислить коэффициенты ускорения выполнения задач и изменения коэффициентов загрузки процессоров в различных режимах. Проанализировать и объяснить полученные результаты.

4. Определить характеристики МВС, обеспечивающие достижение заданного времени выполнения набора задач при минимальных аппаратных затратах (числе процессоров)

$$T_{\text{зад}} = T_{\text{min}} + 4.$$

Варьируемыми параметрами при выполнении этого пункта являются количество процессоров, стратегии назначения готовых к выполнению узлов.

Проанализировать различные варианты изменения характеристик МВС при достижении $T_{\text{зад}}$ также в зависимости от приоритета задач: у задач могут быть равные приоритеты, а также приоритет одной может быть выше приоритета другой.

Исследовать зависимость коэффициента ускорения выполнения набора задач и коэффициентов загрузки процессоров от количества процессоров при различных стратегиях назначения готовых узлов.

Проанализировать полученные результаты, выбрать наилучший способ организации вычислительного процесса.

Отчет должен содержать:

1. Результаты домашней подготовки.
2. Таблицы результатов, полученных в п. 1 – 3 лабораторного задания.
3. Таблицы и графики по п. 3 и 4 лабораторного задания.

Контрольные вопросы

1. Чему равно максимальное и минимальное время выполнения набора задач?
2. Объяснить зависимость коэффициента ускорения от количества процессоров $K_{\text{уск}}(n)$ в однозадачном режиме и при выполнении набора задач.

3. Как влияет учет приоритета задач в наборе на выбор стратегии назначения готовых к выполнению узлов ВП?

4. Объяснить изменения коэффициентов загрузки процессоров при выполнении набора задач и каждой задачи в отдельности.

5. Как определить минимальные аппаратные затраты, обеспечивающие достижение заданного времени выполнения набора задач?

Лабораторная работа № 3

ИССЛЕДОВАНИЕ ПРИНЦИПОВ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В МВС С ОБЩЕЙ ПАМЯТЬЮ

Цель работы: изучение способов организации ВП при выполнении набора задач различных типов на многопроцессорной ВС с общей памятью и шинной организацией коммутации при определении параметров ВС (количество процессоров, шин и модулей памяти), позволяющих выполнить набор задач конкретного типа за заданное время.

Домашняя подготовка

1. Изучить соответствующий раздел лекционного курса и настоящее описание лабораторных работ. Получить у преподавателя исходные данные для выполнения домашней подготовки: граф одной задачи с указанными временами выполнения узлов и передач по шине между узлами.

2. Для заданного графа с использованием алгоритма, приведенного выше, рассчитать критический путь и минимальное время выполнения задачи с учетом времени передач по шине, сравнить полученный критический путь с графом, в котором не учитывается время передач.

3. Выбрать значения параметров структуры МВС с общей памятью, на которой возможно выполнение задачи за минимальное время.

Лабораторное задание

1. Проверить выбранный в п.3 домашней подготовки вариант на модели. Проанализировать и объяснить полученные результаты.

Для выполнения дальнейших пунктов лабораторного задания получить у преподавателя наборы из четырех задач различных типов. В работе исследуются задачи следующих типов:

1) слабосвязанные задачи, в которых время выполнения узлов задачи много больше времени передач между узлами $t_p \gg t_{\Pi}$;

2) среднесвязанные задачи, в которых $t_p \approx t_{\Pi}$;

3) сильносвязанные задачи, в которых $t_p \ll t_{\Pi}$.

2. Построить зависимости времени решения задач от числа процессоров, числа шин и модулей памяти для набора задач каждого типа, найти лучший вариант. Выявить параметры, которые дают наиболее существенный выигрыш.

3. Определить структуру ВС, позволяющую выполнить набор задач каждого типа за заданное время

$$T_{\text{зад}} = 1,33 T_{\text{min}}.$$

Проанализировать полученные результаты и объяснить их.

Отчет должен содержать:

1. Результаты домашней подготовки.
2. Таблицы и графики по п.2 лабораторного задания.
3. Структурные схемы МВС по п.3 лабораторного задания.

Контрольные вопросы

1. Задачи каких типов исследуются в данной лабораторной работе?
2. Как определить критический путь и минимальное время выполнения задачи с учетом времени передач на МВС с общей памятью, какая структура МВС этому соответствует? Всегда ли этот критический путь совпадает с критическим путем задачи без учета передач?
3. Изменение какого параметра (число процессоров, шин или модулей памяти) является наиболее существенным для уменьшения времени выполнения задач различных типов? Как это можно объяснить на основании анализа зависимостей, полученных в п.2 лабораторного задания?
4. Как можно объяснить увеличение в некоторых случаях времени выполнения набора задач при увеличении количества процессоров? На каких

типах задач это наиболее наглядно? Какие значения остальных параметров МВС этому соответствуют?

5. Как определить значения параметров МВС (число процессоров, шин и модулей памяти), позволяющих выполнить набор задач за заданное время при наименьшем количестве шагов моделирования?

Лабораторная работа № 4

ИССЛЕДОВАНИЕ ПРИНЦИПОВ ОРГАНИЗАЦИИ ВП В МВС С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ

Цель работы: изучение способов организации вычислительного процесса в МВС с распределенной памятью для определения параметров МВС (количество процессоров и шин), позволяющих выполнить сильносвязанную задачу за заданное время; сравнение полученных характеристик с аналогичными характеристиками для МВС с общей памятью.

Домашняя подготовка

1. Изучить соответствующий раздел лекционного курса и настоящее описание лабораторных работ. Получить у преподавателя исходные данные для выполнения работы: граф задачи с указанными временами выполнения узлов и передач по шине между узлами.

2. Для заданного графа с использованием алгоритма, приведенного выше, рассчитать минимальное время выполнения задачи в МВС с распределенной памятью. Сравнить полученное значение с минимальным временем выполнения задачи в МВС с общей памятью.

3. Предложить значения параметров структуры МВС с распределенной памятью, на которой возможно выполнение задачи за минимальное время.

Лабораторное задание

1. Проверить выбранный при домашней подготовке вариант на модели, проанализировать и объяснить полученные результаты.

2. Построить зависимости времени решения задачи от числа процессоров и числа шин при решении заданной задачи в МВС с распределенной памятью. Выявить параметры, дающие наиболее существенный выигрыш.

3. Построить аналогичные зависимости для МВС с общей памятью (для сравнения необходимо в п.2 и 3 рассматривать МВС с одинаковыми параметрами).

4. Определить коэффициенты улучшения времени выполнения задачи в МВС с различной организацией; коэффициенты загрузки процессоров и шин в МВС с распределенной памятью и сравнить их с коэффициентами загрузки процессоров и шин в МВС с общей памятью. Проанализировать и объяснить полученные результаты.

Отчет должен содержать:

1. Результаты домашней подготовки.
2. Таблицы и графики по п.2 и 3 лабораторного задания .
3. Таблицу результатов, полученных в п.4 лабораторного задания.

Контрольные вопросы

1. Как определить критический путь и минимальное время выполнения задачи с учетом времени передач на МВС с распределенной памятью? Какая организация МВС (число процессоров, шин, распределение узлов по процессорам) этому соответствует?

2. Как объяснить изменение коэффициентов улучшения времени выполнения задачи на МВС с различной организацией с изменением числа процессоров?

3. Как объяснить изменение коэффициентов загрузки процессоров и шин при выполнении задач на МВС с различной организацией – с распределенной и общей памятью?

4. Объяснить сравнительные характеристики выполнения задач в МВС с общей и распределенной памятью.

1. Амамия М., Танака Ю. Архитектура ЭВМ и искусственный интеллект. М.: Мир, 1993. 400 с.
2. Ладыгин И.И., Белоцицкий Н.С. Основы построения вычислительных систем. М.: Изд-во МЭИ, 1992. 52 с.
3. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. М.: Наука, 1975. 360 с.
4. Бруно Дж. Л., Грэхем Р.Л. и др. Теория расписаний и вычислительные машины. М.: Наука, 1984. 236 с.
5. Барский А.Б. Параллельные процессы в вычислительных системах. Планирование и организация. М.: Радио и связь, 1990. 287 с.