

ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЕГИОНАЛЬНЫЙ ОТКРЫТЫЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Факультет основных образовательных программ
Кафедра информатики, вычислительной техники и автоматизации

ОТЧЕТ
о прохождении производственной практики

Выполнил: студент 3 курса группы ПР19зу
заочной формы обучения
направления подготовки
09.03.04 Программная инженерия
Марчукова Е.В.
(Фамилия И.О.)

Руководитель практики
от предприятия: Ведущий специалист
Дубровина Н.В.
(должность, Фамилия И.О.)

_____ (дата) _____ (подпись)

Руководитель практики
от института: Профессор кафедры
Борзов Д. Б.
(должность, ученая степень, ученое звание, Фамилия И.О.)

_____ (оценка) _____ (дата) _____ (подпись) Курск,

2022

Курск 2023

Оглавление

1. Архитектура массивно-параллельных компьютеров 2
2. Анализ моделей и методов размещения в коммутационном- монтажном проектировании 4
3. Акселератор планирования размещения задач в кластерных вычислительных системах высокой готовности 6

1. Архитектура массивно-параллельных компьютеров

MPP (massive parallel processing) – массивно-параллельная архитектура. Главная особенность такой архитектуры состоит в том, что память физически разделена. В этом случае система строится из отдельных модулей, содержащих процессор, локальный банк операционной памяти (ОП), коммуникационные процессоры (**роутеры**) или сетевые адаптеры, иногда – жесткие диски и/или другие устройства ввода/вывода. По сути, такие модули представляют собой полнофункциональные компьютеры (см. рис. 1).

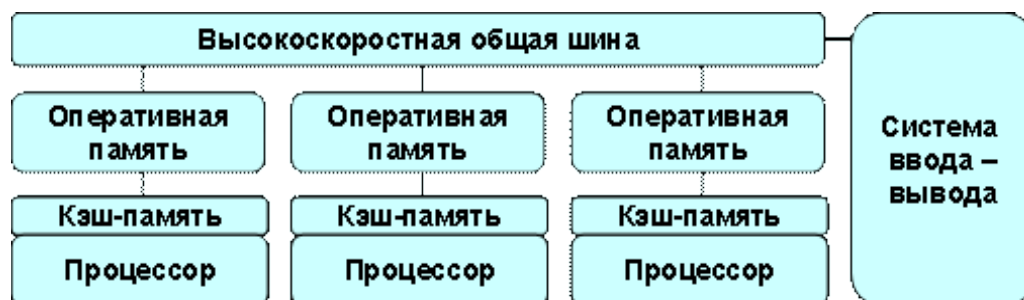


Рисунок 1 – Схематический вид архитектуры с раздельной памятью

Доступ к банку ОП из данного модуля имеют только процессоры (ЦП) из этого же модуля. Модули соединяются специальными коммуникационными каналами. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сообщениями с другими процессорами.

Используются два варианта работы операционной системы на машинах MPP-архитектуры:

- полноценная операционная система (ОС) работает только на управляющей машине (front-end), на каждом отдельном модуле

функционирует сильно урезанный вариант ОС, обеспечивающий работу только расположенной в нем ветви параллельного приложения.

- на каждом модуле работает полноценная UNIX-подобная ОС, устанавливаемая отдельно.

Главным преимуществом систем с раздельной памятью является хорошая масштабируемость: в отличие от SMP-систем, в машинах с раздельной памятью каждый процессор имеет доступ только к своей локальной памяти, в связи с чем не возникает необходимости в потактовой синхронизации процессоров. Практически все рекорды по производительности на сегодня устанавливаются на машинах именно такой архитектуры, состоящих из нескольких тысяч процессоров (ASCI Red, ASCI Blue Pacific).

Недостатки:

- отсутствие общей памяти заметно снижает скорость межпроцессорного обмена, поскольку нет общей среды для хранения данных, предназначенных для обмена между процессорами. Требуется специальная техника программирования для реализации обмена сообщениями между процессорами;

- каждый процессор может использовать только ограниченный объем локального банка памяти;

- вследствие указанных архитектурных недостатков требуются значительные усилия для того, чтобы максимально использовать системные ресурсы. Именно этим определяется высокая цена программного обеспечения для массивно-параллельных систем с раздельной памятью.

Системами с раздельной памятью являются суперкомпьютеры MBC-1000, IBM RS/6000 SP, SGI/CRAY T3E, системы ASCI, Hitachi SR8000, системы Parsytec. Машины последней серии CRAY T3E от SGI, основанные на базе процессоров Dec Alpha 21164 с пиковой производительностью 1200 Мфлопс/с (CRAY T3E-1200), способны масштабироваться до 2048 процессоров.

При работе с MPP-системами используют так называемую Massive Passing Programming Paradigm – парадигму программирования с передачей данных (MPI, PVM, BSPlib).

2. Анализ моделей и методов размещения в коммутационно-монтажном проектировании

Монтажно-коммутационное пространство (МКП) предназначено для размещения конструктивных модулей и трассировки соединений между их контактами, которые должны быть соединены электрическими цепями. Форма и, естественно, математическая модель МКП зависят от уровня модуля, для которого в данный момент решаются задачи конструирования (базовый матричный кристалл, печатная плата, панель и т. д.). В дальнейшем ограничимся только плоским монтажно-коммутационным пространством, соответствующим конструктивному модулю типа печатной платы.

Без потери общности будем считать, что пространство имеет прямоугольную форму, так как введением областей, в которых запрещается размещение конструктивных модулей более низкого уровня или трассировки соединений, можно придать пространству произвольную форму. Так как МКП служит для решения двух задач — размещения модулей и трассировки, — то модели МКП, используемые для решения каждой задачи, будут иметь отличия. Рассмотрим эти модели подробнее.

Наибольшее распространение для решения задач размещения конструктивных модулей в плоском МКП получили эвристические дискретные модели. Такие модели (будем их называть МКП1) строятся следующим образом: МКП разбивается на элементарные площадки (дискреты), каждая из которых предназначена для размещения одного конструктивного модуля более низкого уровня, например микросхемы на печатной плате. Эти площадки в дальнейшем будем называть дискретами рабочего поля (ДРП).

Каждый дискрет в процессе решения задачи размещения может находиться в одном из следующих состояний: свободен для размещения, занят, имеет определенный вес, запрещающий размещение в нем модуля, и т. д. Такая модель МКП отличается простотой и удобством использования в эвристических алгоритмах размещения, однако она не является полностью формализованной.

Одной из разновидностей модели МКП является модель с ортогональной сеткой, в узлах которой могут размещаться модули низкого уровня (рис. 2). Шаг сетки выбирается из условия возможности размещения модулей в соседних узлах сетки.

При размещении разногабаритных компонентов часто размер ДРП выбирают равным наибольшему общему делителю линейных размеров размещаемых модулей либо линейным размерам установочного места для наименьшего из модулей, если размеры всех модулей кратны. Заметим, что выбор шага дискретизации представляется весьма важным, так как при малых размерах ДРП увеличивается время решения задачи, зато повышается плотность заполнения МКП модулями низшего уровня.

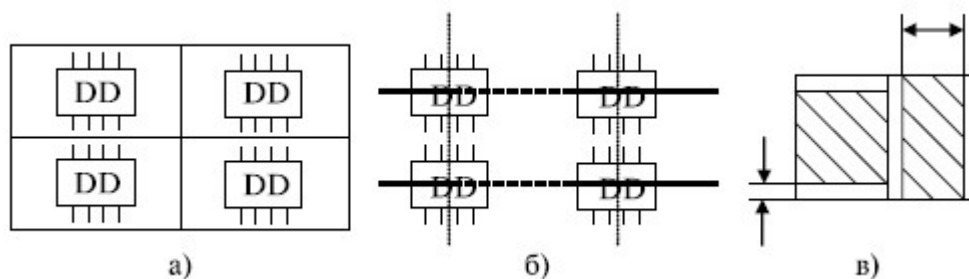


Рис. 2 - Дискретные модели МКП

Аналогичные дискретные модели используются и для решения задач трассировки. В этом случае дискрет является квадратом со сторонами, равными ширине проводника плюс зазор между ними. При этом считается, что проводник из каждого дискрета может быть проведен только в соседний ДРП.

Наибольшее распространение для решения задач размещения получили модели МКП в виде взвешенного графа $VG(S, V)$, которые будем обозначать

МКП2. Взвешенный граф VG представляет собой симметрический граф, в котором множество вершин S соответствует множеству установочных позиций в коммутационном пространстве для модулей низшего уровня, а множество ветвей интерпретирует множество связей между соответствующими установочными позициями. Каждой ветви графа u_{ij} присваивается вес r_{ij} — он равен числу условных единиц расстояния между центрами установочных позиций S_i и S_j , интерпретируемых вершинами, которые инцидентны данной ветви. Вес ветви r_{ij} определяется в зависимости от метрики пространства по одной из формул.

Для описания взвешенного графа VG удобно использовать матрицу смежностей Q , строки и столбцы которой соответствуют вершинам графа, т. е. множеству установочных позиций в МКП, а элементы g_{ij} равны весу ветви, инцидентной i -й и j -й вершинам графа. Элементы, лежащие на главной диагонали матрицы смежностей Q , принимаются равными нулю.

3. Акселератор планирования размещения задач в кластерных вычислительных системах высокой готовности

С началом разработки отказоустойчивых многокомпьютерных и высокодоступных кластеров возрастают требования к скорости выполнения шагов планирования развертывания задач [1-4]. Быстрое восстановление правильного функционирования системы путем отключения неисправного процессора и перенастройки структуры системы путем замены его резервным процессором, обычно не относящимся к процессорам обработки Вашей конфигурация подключения существенно изменится. Формирование длинных маршрутов передачи данных. Их можно уменьшить путем перераспределения оперативных задач.

В то же время этапы планирования развертывания носят комбинаторный характер и отличаются высокой вычислительной сложностью, что может привести к значительному увеличению времени восстановления и снижению коэффициентов готовности системы. По этой

причине не рекомендуется отказываться в переназначении задач до перезагрузки восстановленной системы. Это связано с тем, что увеличение задержек связи может привести к потерям производительности системы, которые превышают ожидаемые выгоды от использования интерактивной многопроцессорной параллельной многопроцессорной обработки. программа.

Следовательно, для сокращения времени восстановления многопроцессорной кластерной системы необходимо значительно сократить время, затрачиваемое на планирование размещения задач по сравнению с программной реализацией на управляющей машине кластера. Этого можно добиться, создав специальный ускоритель. При разработке алгоритмов этой функции целесообразно найти новые способы снижения вычислительной сложности процедуры планирования размещения задач на высокопроцессорных матричных блоках.

В связи с началом освоения отказоустойчивых мультимпьютеров и кластеров высокой готовности повышаются требования к скорости выполнения процедур планирования размещения задач. Быстрое восстановление правильности функционирования системы путем реконфигурации ее структуры с отключением неисправного процессора и заменой его резервным, расположенным обычно вне поля обрабатывающих процессоров, приводит к существенному изменению конфигурации связей между ними и образованию длинных и перекрывающихся маршрутов передачи данных. Они могут быть уменьшены и разнесены путем оперативного перераспределения задач. В то же время процедуры планирования размещения являются комбинаторными, имеют большую вычислительную сложность и поэтому могут привести к существенному увеличению времени восстановления и снижению коэффициента готовности системы. Отказываться из-за этого от перераспределения задач перед рестартом восстанавливаемой системы нецелесообразно, так как возросшие коммуникационные задержки могут привести к такой потере системной

производительности, которая превысит ожидаемый выигрыш от применения параллельной многопроцессорной обработки комплекса взаимодействующих программ. Поэтому для уменьшения времени восстановления многопроцессорных кластерных систем необходимо многократно снизить затраты времени на планирование размещения задач. Этого можно достичь путем создания специализированного ускоряющего вычислительного устройства (акселератора), а при разработке алгоритмов его функционирования целесообразно найти новый метод снижения вычислительной сложности процедур планирования размещения задач по процессорам матричных базовых блоков кластерных систем высокой готовности.

В связи с вышеизложенным актуальной является научно-техническая задача многократного повышения скорости выполнения процедур планирования размещения параллельно обрабатываемых задач по процессорам кластерной системы путем реализации названных процедур в специализированном вычислительном устройстве.

Разработанный метод ускорения поиска субоптимального варианта размещения задач по процессорам базового матричного кластерного блока основан на следующем подходе.

Пакет взаимодействующих программ (задач), запланированных к обработке в базовом блоке, описывается графом взаимодействия задач

$$G = \langle X, E \rangle, \text{ где } X = \left\{ \begin{matrix} x_{1,1} & x_{1,2} & \dots & x_{1,k} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,k} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{q,1} & x_{q,2} & \dots & x_{q,k} & \dots & x_{q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & x_{n,2} & \dots & x_{n,k} & \dots & x_{n,n} \end{matrix} \right\} \text{ а - аа (1)}$$

множество вершин графа G , вершины $x_{qk} \in X$ которого соответствуют задачам, а дуги связей между ними $e_{ij} \in E$ априори $i, j = (q-1) \cdot n + k$ авзвешиваются объемами данных M_{ij} , передаваемыми

между задачами и сведенными в матрицу обмена информацией (МОИ) $M = \|m_{ij}\|_{N \times N}$, где $N = n^2 = |X|$.

Матричный базовый блок кластерной системы представляется топологической моделью в виде графа $H = \langle P, V \rangle$,

$$P = \begin{Bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,n} \\ \dots & \dots & \dots & \dots \\ P_{n,1} & P_{n,2} & \dots & P_{n,n} \end{Bmatrix}$$

где a - множество идентификатора процессорных модулей базового блока, организованных в матрицу $|P|n^n$, где $|P| = N = n^2$ - число процессорных модулей базового блока; V - множество межмодульных связей, задаваемых матрицей смежности $\|W\|_{N \times N}$ размером $n^2 \times n^2$.

Размещение пакета программ (задач), описываемых графом G (1), в параллельной системе (ПС) может быть аналитически описано отображением

$$\beta_s = \begin{Bmatrix} x_{s_1,1} & x_{s_1,2} & \dots & x_{s_1,k} & \dots & x_{s_1,n} \\ x_{s_2,1} & x_{s_2,2} & \dots & x_{s_2,k} & \dots & x_{s_2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{s_q,1} & x_{s_q,2} & \dots & x_{s_q,k} & \dots & x_{s_q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{s_n,1} & x_{s_n,2} & \dots & x_{s_n,k} & \dots & x_{s_n,n} \end{Bmatrix} \rightarrow \begin{Bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,k} & \dots & P_{1,n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,k} & \dots & P_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{q,1} & P_{q,2} & \dots & P_{q,k} & \dots & P_{q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{n,1} & P_{n,2} & \dots & P_{n,k} & \dots & P_{n,n} \end{Bmatrix}, \quad (2)$$

где $s = \overline{1, N!}$, $k = \overline{1, n}$, $q = \overline{1, n}$.

Здесь S_a - это номер очередной перестановки задач $\{x_{qk}\}$ по процессорным модулям $\{P_{qk}\}$, соответствующий S -му варианту размещения. Мощность множества $\psi = \{\beta_s\}$ всех возможных отображений (2) равна числу всевозможных перестановок задач $\{x_{qk}\}$ в матрице X : $|\psi| = N!$. Для описания множества длин d_{ij} кратчайших маршрутов передачи данных в пределах

базового блока введем матрицу минимальных расстояний (ММР) $D = \|d_{ij}\|_{N \times N}$, $N = n^2 = |P|$, которая строится по матрице смежности.

Задачу планирования размещения, можно сформулировать как поиск такого отображения $b^* \in \Psi$, что

$$T_{\beta^*} = \min_{\psi} \left\{ \max_{\beta_s \in \psi} \left\{ T_{\beta_s} (p_{a,b}, p_{x,y}) \right\} \right\}, \quad (3)$$

где $T_{\beta_s} (p_{a,b}, p_{x,y})$ - коммуникационная задержка при передаче данных между процессорными модулями $p_{a,b}$ и $p_{x,y}$, соответствующая отображению β_s вычисляемая как произведение

$$T_{\beta_s} (p_{a,b}, p_{x,y}) = d_{ij} \cdot m_{ij} \cdot C, \quad (4)$$

$$\text{где } i = (a-1) \cdot n + b \text{ и } j = (x-1) \cdot n + y,$$

$$\max_{\beta_s \in \psi} \left\{ T_{\beta_s} (p_{a,b}, p_{x,y}) \right\} = \max_{aa} \left\{ d_{ij} \cdot m_{ij} \right\}. \quad (5)$$

Присутствующий в (4) множитель C не учитывается в выражениях (3,5) и последующем анализе, так как он обратно пропорционален постоянной скорости передачи данных и поэтому не влияет на результаты минимизации по (3).

Поиск наилучшего варианта размещения b^* по критерию (3) является сложной переборной задачей. Одним из путей его ускорения может быть применение целенаправленных перестановок строк и столбцов матрицы МОИ с выбором в ней ak -го места перестановки ее элемента $m_{\alpha,\beta}$ по критерию:

$$d_{\alpha k} < d_{\alpha\beta}, \quad \text{aa} \quad (6)$$

где $d_{\alpha k}, d_{\alpha\beta}$ - одноименные элементы матрицы ММР; $m_{\alpha,\beta}$ - элемент МОИ, которому соответствует $\max \{m_{ij} \cdot d_{ij}\}$, найденный в предыдущем шаге перестановок.

Новизна подхода состоит в том, что общее число требуемых перестановок можно дополнительно уменьшить, если отбросить явно нецелесообразные из них, разрешая очередную перестановку по следующим дополнительным критериям:

$$m_{\alpha k} \cdot d_{\alpha k} < m_{\alpha \beta} \cdot d_{\alpha \beta}, \text{aa} \quad (7)$$

$$m_{\alpha k} < m_{\alpha \beta}, \text{aa} \quad (8)$$

Многokратное ускорение поиска возможно за счет допустимого снижения выигрыша на снижение величины коммуникационной задержки, например, не более, чем на 20-30%, по сравнению с лучшими результатами, достигаемыми при больших затратах времени на поиск. Для этого необходимо в ходе поиска контролировать степень уменьшения величины образующейся коммуникационной задержки (3) и принимать решение о целесообразности продолжения поисковых перестановок строк и столбцов матрицы МОИ. Процедура принятия решения основана на вычислении недостижимой минимальной оценки размещения T_{inf} (гипотетического минимума коммуникационной задержки) при допущении, что топологии графов G и H тождественны. При вычислении нижней оценки будем назначать дуги графа G с наибольшим весом m_{ij} ана самые короткие маршруты в графе H , не обращая внимания на ограничения, накладываемые фактическими связями между задачами в графе G . Соответствующий формализованный алгоритм выглядит следующим образом.

- Переписать элементы $d_{kl} \neq 0$ матрицы D в вектор-строку $D' = \|d_{kl}^z\|$ так, что $d_{kl}^{z_1} \leq d_{kl}^{z_2} \Leftrightarrow z_1 > z_2$, где z_1 и z_2 - порядковые номера элементов в D' .

- Переписать элементы $m_{ij} \neq 0$ матрицы M в вектор-строку $M' = \|m_{ij}^z\|$ так, что $m_{ij}^{z_1} \geq m_{ij}^{z_2} \Leftrightarrow z_1 > z_2$, где z_1 и z_2 - порядковые номера элементов в M' .

– Положить

$$T_{\text{inf}} = \max\{m^z d^z\},$$

где $z = 1, \overline{|E|}$, а $|E|$ - мощность множества E , равная количеству дуг в графе G ; m^z, d^z - одноименные элементы векторов \mathbf{M} и \mathbf{D} с одинаковыми порядковыми номерами от начала названных выше вектор-строк.

Для многократного ускорения поиска разработана следующая методика ускоренного выполнения процедур планирования размещения задач.

1. Составляются две матрицы: обмена информацией между задачами (МОИ) и кратчайших маршрутов (ММР) между процессорами в коммуникационной среде базового блока.

2. Вычисляются гипотетический минимум коммуникационной задержки T_{inf} и коэффициент эффективности исходного произвольного размещения задач $\eta_{\text{н}} = T_{\text{н}} / T_{\text{inf}}$.

3. По порогу эффективности $\eta_{\text{н}} > 2$ принимается решение о целесообразности инициализации процедуры поиска субоптимального размещения. Под коэффициентом эффективности перестановок $\eta = T / T_{\text{inf}}$ понимается отношение реально полученной величины задержки (5) к гипотетической T_{inf} .

4. Выполняются шаги целенаправленных перестановок столбцов и строк матрицы обмена информацией. Находится максимальное значение коммуникационной задержки (5) по предыдущему варианту перестановок задач.

5. Находится минимум (3) из максимумов задержек по всем вариантам перестановок и вычисляется коэффициент эффективности η .

6. Если η оказывается менее установленного порога эффективности $\eta \leq 2$, шаги поиска прекращаются и найденный вариант матрицы обмена информацией считается соответствующим субоптимальному размещению.

Величина порога эффективности $\eta_{\Pi} = 2$, выбрана в результате статических исследований на программной модели алгоритма функционирования АПР.

На основании разработанных в данной главе метода ускорения поиска и методики ускорения выполнения процедур планирования размещения составлен следующий алгоритм, программно-аппаратно реализованный в двухуровневом микропроцессорном акселераторе.

– Ввести $M = \|m_{ij}\|_{N \times N}$, $D = \|d_{ij}\|_{N \times N}$, $M1 = \|m1_{ij}\|_{N \times N}$, $M2 = \|m2_{ij}\|_{N \times N}$, $M3 = \|m3_{ij}\|_{N \times N}$, $i = \overline{1, N}$, $j = \overline{1, N}$ "mij=0, "m1ij=0, "m2ij=0, "m3ij=0.

2. Переписать "mij в $\overline{M} = \left\| \vec{m}_{ij} \right\|$ с изменением порядка следования элементов так, что $m_{ij}^{z_1} \geq m_{ij}^{z_2} \Leftrightarrow z_1 < z_2$, где z_1 и z_2 порядковые номера элементов \overline{M} .

3. Переписать "dij в $\overline{D} = \left\| \vec{d}_{ij} \right\|$ с изменением порядка следования элементов так, что $d_{ij}^{z_1} \leq d_{ij}^{z_2} \Leftrightarrow z_1 < z_2$, где z_1 и z_2 порядковые номера элементов \overline{D} .

Положить $T_{\text{inf}} = \max\{m^z d^z\}$, где $z = \overline{1, |E|}$, $E \leq N^2 - N$.

5. Найти $T_N = \max\{m_{ij} \cdot d_{ij}\}$, $j = \overline{1, N}$, где N - число задач.

6. Вычислить $\eta_n = \frac{T_n}{T_{\text{inf}}}$. Если $\eta < \eta_{\Pi}$, то останов, иначе перейти к п.7.

7. Принять $T_0 = T_n$.

8. Выполнить $M2 = M: \forall (i, j) \Rightarrow M2_{ij} = M_{ij}$.

9. Принять $k=1$.

10. Выбрать $m_{k, \text{из}} \overline{M}$.

11. Найти $m_{\alpha\beta}^2$ из $M2$ такой, что $m_{\alpha\beta}^2 = m_k$, и - соответствующий ему $d_{\alpha\beta}^1$ из D .

12. Если $d_{\alpha\beta}^1 = 1$, то $M2_{\alpha\beta} = -1$, $k=k+1$ и перейти к п.10, иначе п.13;

13. Принять $i=1$.

14. Если $i \leq 64$, то перейти к п. 15, иначе - п.25.

15. Если $d_{\alpha i} < S$ и $d_{\alpha i} \neq 0$ и $m_{\alpha\beta} \cdot d_{\alpha\beta} > m_{\alpha i} \cdot d_{\alpha i}$ и $m_{\alpha\beta} \cdot d_{\alpha\beta} > m_{\alpha i} \cdot d_{\alpha\beta}$, то перейти к п.16, иначе $i=i+1$ и - п.14.

16. Для M выполнить операцию $i \leftrightarrow \beta$ и сформировать $M1$:

$$\forall((l, j \neq i) \& (l, j \neq \beta)) \Rightarrow M1_{ij} = M_{ij}, \forall((l = i) \& (j \neq \beta)) \Rightarrow M1_{ij} = M_{j\beta},$$

$$\forall((l = \beta) \& (j \neq i)) \Rightarrow M1_{ij} = M_{ij}, \forall((l \neq i) \& (j = \beta)) \Rightarrow M1_{ij} = M_{ik},$$

$$\forall((l \neq \beta) \& (j = i)) \Rightarrow M1_{ij} = M_{i\beta}, M1_{i\beta} = M_{j\beta}, M1_{j\beta} = M_{i\beta}.$$

17. Вычислить $T_1 = \max\{m_{ij} \cdot d_{ij}\}$. Если $T_1 \leq T_0$, то перейти к п.18, иначе $i = i + 1$ и - п.14.

18. Вычислить $\eta = \frac{T_k}{T_1}$. Если $\eta < \eta_{\text{П}}$, то останов и выдача $M1$, иначе перейти к п.19.

19. Принять $M = M1$: $\forall(l, j) \Rightarrow M_{ij} = M1_{ij}$.

20. Принять $T_0 = T_1$.

21. Принять $M2_{\alpha\beta} = -1$.

22. Для $M2$ выполнить операцию $i \leftrightarrow \beta$ и сформировать $M3$:

$$\forall((l, j \neq i) \& (l, j \neq \beta)) \Rightarrow M3_{ij} = M2_{ij}, \forall((l = i) \& (j \neq \beta)) \Rightarrow M3_{ij} = M2_{j\beta},$$

$$\forall((l = \beta) \& (j \neq i)) \Rightarrow M3_{ij} = M2_{ij}, \forall((l \neq i) \& (j = \beta)) \Rightarrow M3_{ij} = M2_{ik},$$

$$\forall((l \neq \beta) \& (j = i)) \Rightarrow M3_{ij} = M2_{i\beta}, M3_{i\beta} = M2_{j\beta}, M3_{j\beta} = M2_{i\beta}.$$

23. Принять $M2 = M3$: $\forall(l, j) \Rightarrow M2_{ij} = M3_{ij}$.

24. Принять $k=k+1$ и перейти к п.26.

25. $M_{\text{эф}}^k = -1, k=k+1.$

26. Если $k < |M|$, то останов и выдача M1, иначе перейти к п.9

В третьей главе описаны программная модель разработанного алгоритма планирования размещения задач и результаты статистических исследований его эффективности.

Для моделирования и тестирования разработанного метода планирования размещения задач в кластерных системах была разработана на языке Си++ программная система, которая позволяет программно реализовать алгоритм размещения, строить графики изменения показателей

эффективности $\eta = \frac{T}{T_{\text{нр}}}$ и $\sigma = \frac{T_{\text{нр}}}{T}$ при каждой пробной перестановке и с заданной точностью фиксировать время расчета.

Целью исследования было определение величины выигрыша Δ в снижении задержки в результате применения разработанного метода при разных видах и степенях заполнения МОИ, соответствующих широкому диапазону изменения степени связности между задачами пакета программ, запланированных к обработке в базовом матричном блоке. Результаты, соответствующие матрице наилучшего размещения M_k , начальному и достигнутому отклонению задержки T от T_{inf} : $\eta_{\text{наи}}$ и $\sigma_{\text{соот}}$ соответственно, достигнутому выигрышу s в разгах, а так же времени, затраченному на поиск, выводятся на экран.