

Моделирование реконфигурируемой системы реального времени с учетом уменьшения временных затрат обработки информации

Борзов Д.Б.^{1*}, Кошелев М.А.^{1}, Соколова Ю.В.^{2***}**

¹*Юго-Западный государственный университет, ЮЗГУ,*

ул. 50 лет Октября, 94, Курск, 305040, Россия

²*Научно-производственное объединение им. С.А. Лавочкина,*

ул. Ленинградская, 24, Химки, Московская область, 141400, Россия

**e-mail: borzovdb@kursknet.ru*

***e-mail: maks46.krutoe@gmail.com*

****e-mail: jv.sokolova@mail.ru*

Статья поступила 25.02.2021

Аннотация

Статья посвящена исследованиям, способствующим повышению быстродействия реконфигурируемой системы реального времени, путём создания методики и алгоритма распределения массива вычислительных задач. Методы исследования работы основаны на определениях теории множеств, графов, теории вероятности и математической статистики. С их помощью была создана математическая модель и система критериев реконфигурируемой вычислительной системы реального времени, построенной на беспроводном протоколе, позволяющая выполнять размещение задач по мин-максной оценке времени. А также предложен алгоритм размещения задач в вычислительном комплексе, позволяющий увеличить производительность системы за счёт сокращения времени передачи данных. В результате полученная система дала выигрыш по времени примерно в 2,5 раза. Полученная модель может найти применение в распределённых вычислениях.

Ключевые слова: алгоритм, беспроводной протокол, многопроцессорная система, реконфигурируемая вычислительная система.

Постановка задачи

За последние 10 лет ИТ-сфера совершила рывок вперед. Он развивается, затрагивая все больше и больше областей науки и техники. И одним из необходимых аргументов в этой области является создание системы, способной адаптироваться к разным типам решаемых задач и различным областям применения на практике от гражданского до военного. Это один из перспективных способов повышения реальной производительности всего ИТ-комплекса [1].

С развитием беспроводных протоколов были начаты и разработаны реконфигурируемые компьютерные системы, состоящие из нескольких процессорных модулей [2]. Они могут обрабатывать данные независимо, сокращая время, необходимое для выполнения общей задачи, и тем самым повышая производительность всего комплекса в целом [3]. За счет регулярной замены обоих процессорных модулей и их расположения снижается вероятность утечки данных. Это позволяет применять их и в авиационной и ракетно-космической технике. Хост-устройство также меняется, что затрудняет кибератаки [4]. Отказ любого из процессорных модулей никоим образом не повлияет на работу всей системы, поскольку возложенная на него задача все равно будет завершена после ее распределения [5].

Исходя из вышесказанного, беспроводные протоколы позволяют построить реконфигурируемую защищенную информацию отказоустойчивую систему,

элементы которой могут быть распределены по большой площади. Такие системы используются во многих сферах деятельности, от промышленности до военного комплекса [6]. В связи с этим моделирование беспроводной передачи данных с возможностью перемещения объекта приема-передачи является актуальной задачей.

Метод решения проблемы

Для построения реконфигурируемой вычислительной системы реального времени необходимо выполнить ряд действий, направленных на составление матрицы расстояний $A_{N \times N}$ и выбор мастера. Матрица расстояний содержит информацию о расстоянии между процессорными модулями и ведущим, который отвечает за запрос и позиционирование задач для процессорных модулей, входящих в систему в текущий момент.

Мастера выбирают по условиям:

$$1) r = \min \{ \sum N r_{ij} \},$$

где N - количество процессорных модулей, r_{ij} - значения матрицы расстояний,

$$2) r_n = \text{count} (r_{\max}),$$

где r_{\max} - максимум на строку. В условии $r_m < r_n$.

$$3) G_m > G_{PM},$$

где G_m - усиление антенны ведущего, G_{PM} - усиление антенны процессорного модуля, претендующего на роль ведущего.

Условие 1 необходимо и достаточно, условия 2 и 3 используются как дополнительные, если первое условие не раскрыло мастера.

В заданных условиях был составлен алгоритм выбора мастера из набора устройств, представленный на рисунке 1.

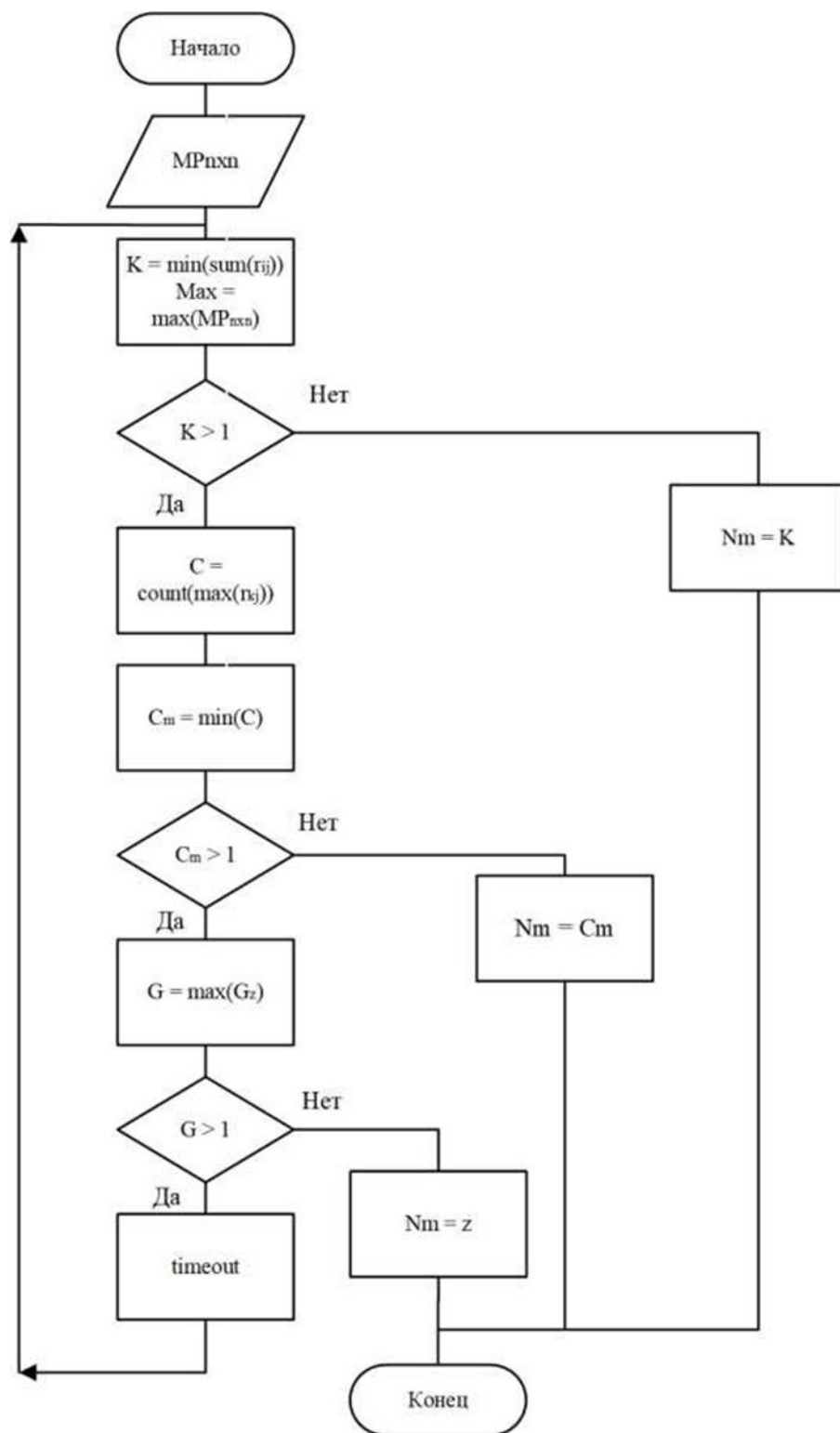


Рисунок 1 – Алгоритм выбора мастера

Алгоритм, представленный на рисунке 1, состоит из нескольких шагов:

1. Для каждой строки матрицы расстояний вычисляется сумма. Вычисляется максимум во всей матрице расстояний.

2. Из полученных сумм выбирается минимум. При наличии нескольких минимальных сумм, то есть на роль мастера несколько кандидатов, выполняется следующее условие.

3. Среди устройств с суммой минимальных строк в матрице расстояний учитывается количество максимумов, найденных на первом этапе. Этот показатель сравнивается. Тот, у кого меньше максимумов, становится мастером. Если и при этом мастер не открылся, выполняется 4 шага.

4. Для тех устройств, у которых в ряду минимум максимумов, сравнивается усиление антенны. Устройство с наивысшим коэффициентом становится мастером.

Чтобы заполнить матрицу расстояний, вам необходимо знать уровень сигнала. Этот показатель вместе с физическим расстоянием влияет на скорость передачи данных и рассчитывается при подключении устройства к точке доступа [7,8]. В свою очередь, расчет расстояния между устройствами - процесс трудоемкий и не дает точного результата (десятки метров). Следовательно, коэффициент расстояния будет записан в матрице расстояний, как написано выше, а не физическое расстояние [9,10].

Таким образом, методология создания реконфигурируемой вычислительной системы реального времени состоит из следующих шагов:

1. Во время ожидания система фиксирует количество активных устройств.
2. Матрица расстояний составляется из коэффициентов расстояния.
3. Мастер выбран.

Стоит отметить, что процессорные модули работают в режиме точки доступа.

Каждый активный процессорный модуль Wi-Fi начинает строить реконфигурируемую вычислительную систему в реальном времени.

На первом этапе каждому устройству, которое подключается к системе, присваивается идентификатор, уникальный в рамках текущей системы [11].

Во время ожидания процессорные модули обмениваются служебными пакетами, содержащими идентификаторы процессорного модуля, усиление антенны и различную информацию [12].

Параллельно с вычислениями процессорные модули продолжают обмениваться пакетами сервисного обмена, данные матрицы расстояний обновляются и сравниваются с текущими. Если мастер изменяется во время изменений, текущий мастер завершает начатые операции, отправляет задачи источнику, если ему удалось их запросить, и продолжает действовать как обычный модуль процессора, в то время как новый мастер уже выполняет свои функции [13].

В соответствии с описанным выше методом был составлен алгоритм построения реконфигурируемой вычислительной системы реального времени. Он работает в двух взаимосвязанных потоках [14,15]. Благодаря этому все изменения отслеживаются, и реакция на них происходит в режиме реального времени.

После построения системы процессорный модуль, назначенный мастером, начинает выполнение возложенной на него задачи – размещения полученных заданий на процессорных модулях комплекса [16,17].

Для выполнения этой функции предлагается процедура с критерием оптимальности распределения равном величине отклонения мин-максной оценки времени пересылки пакетов всех задач от ее нижней оценки K_{y_min} .

Нижняя оценка времени – это минимально возможное время передачи задания в текущей системе [18].

Нижняя оценка времени – недостижимое значение в силу того, что рассылка заданий производится мастером, и расстояния до каждого процессорного модуля фиксированы. При расчете K_{y_min} выбираются минимальные значения расстояний из всего множества значений. То есть поиск нижней оценки времени передачи сводится к следующему:

1. Отсортировать объемы заданий по убыванию.
2. Записать в \vec{A} значения коэффициентов расстояний в порядке возрастания.
3. Исключить из \vec{A} все нули.
4. Перевести $\vec{A} [0]$ в Мб/с.

Найти K_{y_min} , как частное максимального объёма данных на полученное значение скорости передачи данных.

Полученное значение – минимальное время передачи данных.

Для выполнения размещения, необходимо учитывать расстояния, доступные мастеру и возможные цепочки пересылок через другие процессорные модули, способные сократить время [19,20]. Поэтому методика размещения состоит из следующих этапов:

Этап 1: Вычисляется оценка времени пересылки первоначального размещения путем наложения МОЗ $V = \|v_{ij}\|_{N \times N}$ на МР $A = \|a_{ij}\|_{N \times N}$ по формуле:

$$K_{y_t} = \max \left(\frac{v_{ij}}{a_{ij}} \right) \quad (1)$$

где a_{ij} – переведённый в скорость передачи данных коэффициент расстояния.

Этап 2: Вычисляется нижняя оценка времени пересылки по следующему алгоритму:

1. Задания переписываются из МОЗ в вектор в порядке убывания:

$\forall v_{ij}$ в $\vec{V} = \|v_{ij}\|$ так что $v_{ij}^{z_1} \geq v_{ij}^{z_2} \Leftrightarrow z_1 < z_2$, где z_1, z_2 – порядковый номер элемента \vec{V} .

2. Доступные значения КР заносятся из МР в вектор в порядке возрастания:

$\forall v_{ij}$ в $\vec{A} = \|a_{ij}\|$ так что $a_{ij}^{z_1} \leq a_{ij}^{z_2} \Leftrightarrow z_1 < z_2$, где z_1, z_2 – порядковый номер элемента \vec{A} .

3. Удалить из вектора $\vec{A} \forall a_{ij} = 0$.

4. Выбрать первый элемент вектора \vec{A} : $C = \vec{A} [0]$.

5. Перевести C в Мб/с.

6. Рассчитать $K_{y_min} = \frac{\vec{V} [0]}{C}$

Этап 3: Выполняется размещение задач так что $v_{z_1} \geq v_{z_2}$ и $a_{z_1} \leq a_{z_2}$.

Этап 4: Вычисляется время пересылки:

$$K_{y_H} = \max \left(\frac{v_{mj}}{a_{mj}} \right) + \frac{v_{ij}}{a_{ij}}, \quad (2)$$

где m – номер мастера, $a_{ij} \neq 0$.

Этап 5: Анализ величины η по формулам:

$$\eta_{H_min} = \frac{K_{y_H}}{K_{y_min}};$$

$$\eta_{T_min} = \frac{K_{y_T}}{K_{y_min}};$$

$$\eta_w = \frac{\eta_{T_min}}{\eta_{H_min}}.$$

Коэффициент приближения η_{n_min} – показывает во сколько раз размещение приблизило время пересылки к минимуму.

Коэффициент начального размещения η_{r_min} , означающий во сколько раз начальное размещение хуже минимального.

Выигрыш η_w – минимаксный критерий, указывающий во сколько раз новое размещение выгоднее изначального.

По выделенным этапам размещения составлен следующий алгоритм:

1. Записать в вектор доступные мастеру КР: $\forall a_{ij}$ в $\vec{A} = \| a_{ij}^z \|$ где z – порядковый номер элемента \vec{A} .
2. Переписать выбранные значения в другой вектор в порядке возрастания: $\forall a_{ij}$ в $\overrightarrow{A_{min-max}} = \| a_{ij}^z \|$ так что $a_{ij}^{z_1} \leq a_{ij}^{z_2} \Leftrightarrow z_1 < z_2$, где z_1, z_2 – порядковый номер элемента $\overrightarrow{A_{min-max}}$.
3. Получить задания: $\forall a_{ij}$ в $\vec{V} = \| v_{ij}^z \|$ где z_1 – порядковый номер элемента \vec{V} .
4. Вычислить необходимые для размещения значения: $V_{cp} = \frac{\sum \vec{A}}{N}$, где N – размер \vec{A} . $V_{max} = \max(\vec{A})$. $K_{y_r} = \max\left(\frac{v_{ij}}{a_{ij}}\right)$.
5. Получить минимальное время пересылки задания: $K_{y_min} = \frac{\vec{V}[0]}{c}$.
6. Составить вектор минимумов: $\overrightarrow{V_{min}} = \| v_{ij}^z \|$ так что $v_{ij}^{z_1} \leq V_{cp} \Leftrightarrow z_1 < z_2$, где z_1, z_2 – порядковый номер элемента $\overrightarrow{V_{min}}$.
7. Записать массив максимумов: $\overrightarrow{V_{max}} = \| v_{ij}^z \|$ так что $v_{ij}^{z_1} \geq V_{cp} \Leftrightarrow z_1 < z_2$, где z_1, z_2 – порядковый номер элемента $\overrightarrow{V_{max}}$.

8. Отсортировать $\overrightarrow{V_{min}}$ и $\overrightarrow{V_{max}}$ в порядке убывания: $v_{z1} > v_{z2}$, где z_1, z_2 – порядковый номер элемента $\overrightarrow{V_{min}}$ и $\overrightarrow{V_{max}}$.
9. Составить пары связанных задач $\vec{L} = v_{ij} \div v_{ij}$ так что $p_{ij} = 1$.
10. Вычислить возможные суммы при: $l_z \neq V_{max}$; $l_{z1} \in \overrightarrow{V_{min}}$ $\frac{\text{или}}{\text{и}}$ $l_{z2} \in \overrightarrow{V_{min}}$; $l_{z1} + l_{z2} > V_{cp}$.
11. Записать суммы в вектор $\overrightarrow{V_1}$. Исключить $l_z = \min(l_{z1}, l_{z2})$ из $\overrightarrow{V_{min}}$. Записать $l_z = \min(l_{z1}, l_{z2}) \in \overrightarrow{V_{min}}$ в $\overrightarrow{V_2}$.
12. Создать матрицу $M_p = |a_{ij}|_{N \times N}$.
13. Распределить $m_{NMNM} \leftarrow V_{max}$. В \vec{A} записать $a_{NM} = -1$.
14. Отсортировать $\overrightarrow{V_1}$ и $\overrightarrow{V_2}$ так что $v_{z1} > v_{z2}$, где z_1, z_2 – порядковый номер элемента $\overrightarrow{V_1}$ и $\overrightarrow{V_2}$.
15. Разместить $\overrightarrow{V_1}$ так что $m_{ij} \leftarrow v_z$. Записать в $\overrightarrow{A_{min-max}} a_z = -1$.
16. В \vec{A} записать $a_{NM} = -1$ при $a_z = a_{mmz}$, где a_{mmz} – элемент $\overrightarrow{A_{min-max}}$, $i, j = z$.
17. Распределить $\overrightarrow{V_2}$.
- 17.1 Выписать $\overrightarrow{A_K} = \|a_{ij}^z\|$ при $v_z = m_{pij}$ $k = i$.
- 17.2 Сравнить вектора и приравнять: $\overrightarrow{A_K} = \vec{A}$ для элементов $a_z = -1$.
- 17.3 Найти минимум из значений вектора $\overrightarrow{A_K}$: $z = \min(\overrightarrow{A_K})$, $m_{zz} \leftarrow v_z$, где v_z – элемент $\overrightarrow{V_2}$, m_{zz} – элемент $M_{pN \times N}$.
18. Разместить $\overrightarrow{V_{min}}$ так что $m_{ij} \leftarrow v_z$, где $v_{z1} > v_{z2}$ и $a_{z1} < a_{z1}$.
19. Вычислить: $K_{y-n} = \max\left(\frac{v_{mj}}{a_{mj}}\right) + \frac{v_{ij}}{a_{ij}}$, где $a_{ij} \neq 0$.

20. Посчитать коэффициенты: $\eta_{H_min} = \frac{K_{y_H}}{K_{y_min}}$; $\eta_{T_min} = \frac{K_{y_T}}{K_{y_min}}$; $\eta_w = \frac{\eta_{T_min}}{\eta_{H_min}}$.

Теоретические расчеты показали, что алгоритм размещения позволяет уменьшить время пересылки заданий в 2 раза. Для дальнейшего анализа и сравнения с аналогами предложенный алгоритм будет промоделирован в следующем разделе.

Результаты и их обсуждение

Для моделирования и тестирования разработанных алгоритмов построения реконфигурируемой системы реального времени и размещения задач в ней на языке C++ разработана программа, позволяющая моделировать алгоритм размещения и построения системы. Общий алгоритм моделирования представлен на рисунке 2. Он состоит из блоков ввода начальных данных и вывода результатов, двух блоков основных взаимосвязанных процессов – функции `surv()` и `allocstion()` и блоков условий, контролирующих время.

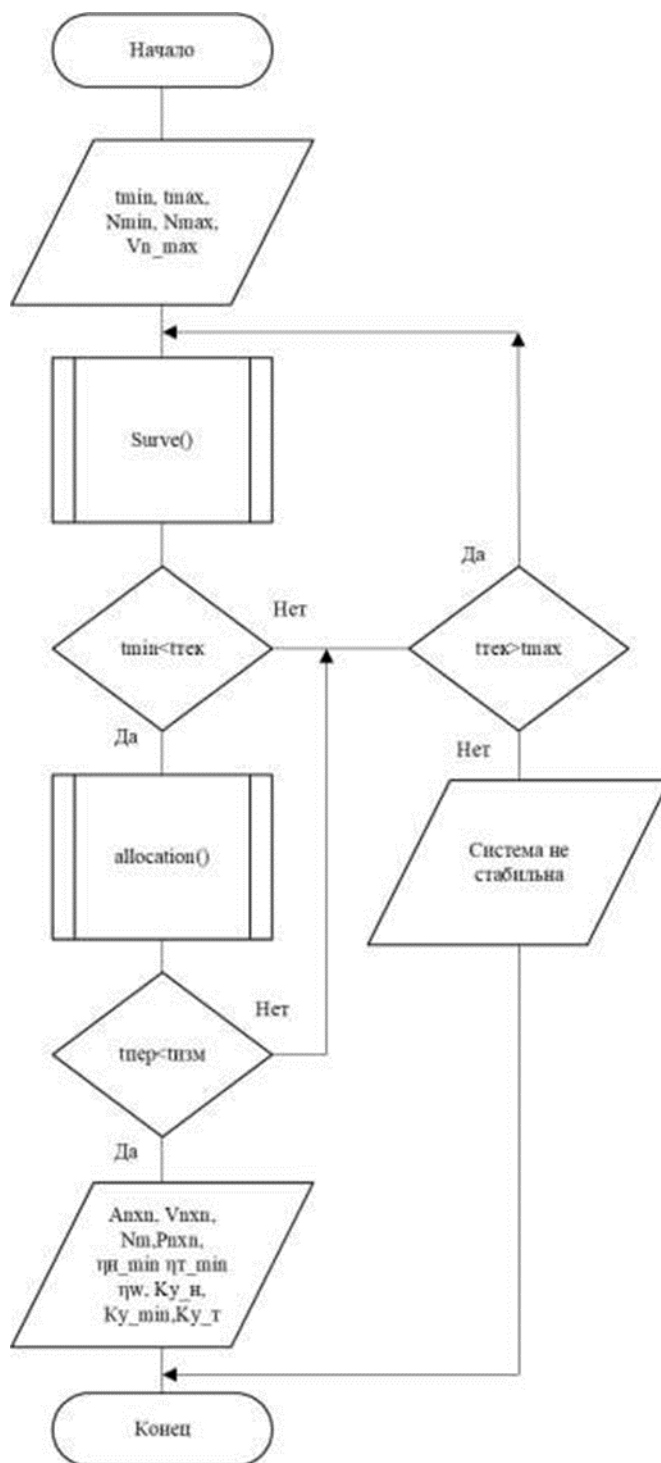


Рисунок 2 – Общий алгоритм программы

В размещении показаны матрицы расстояний и связей, не меняющиеся во время размещения и матрица объёмов заданий до и после размещения. В показателях выводятся данные, необходимые для анализа алгоритма размещения.

Программа обладает следующими функциональными возможностями: ввод времени ожидания, задание минимального и максимального числа процессорных модулей в системе, ограничение максимального объёма передаваемых данных, схематический вывод состояний систем с текущими данными. После выполнения алгоритма распределения программа предоставляет данные посчитанные в процессе выполнения минимальное K_{y_min} и максимальное K_{y_max} время передачи задания, время до размещения $K_{y_т}$, и после него $K_{y_н}$, номер мастера N_m , выигрыш от алгоритма размещения η_w .

Для анализа работы алгоритма размещения в реконфигурируемой вычислительной системе реального времени необходимо выполнить моделирование и изучить разные зависимости. В данной работе будут рассмотрены следующие зависимости:

1. Зависимость скорости выполнения алгоритма, выраженная в тактах, относительно количества процессорных модулей;
2. Зависимость выигрыша от количества связанных задач.

Выигрыш η_w – мин-максный критерий, указывающий во сколько раз новое размещение выгоднее изначального.

Для рассмотрения пункта 1 было подсчитано время выполнения алгоритма размещения в тактах процессора. Результаты представлены на рисунке 3.

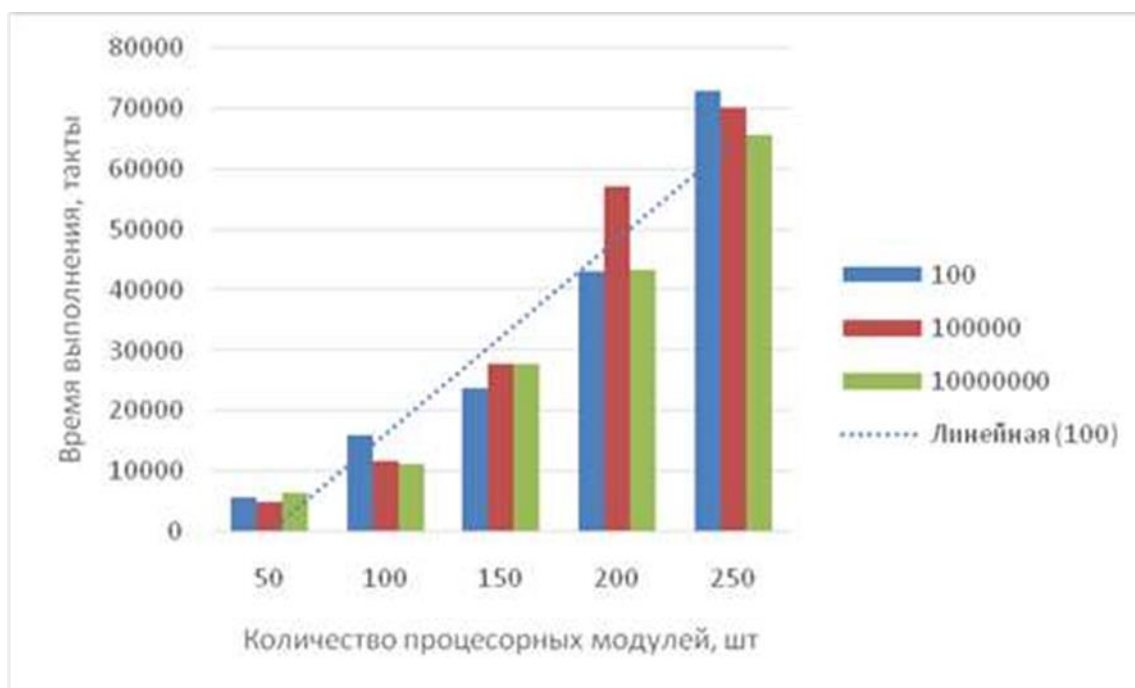


Рисунок 3 – Зависимость времени выполнения алгоритма от количества процессорных модулей.

На рисунке 3 по оси абсцисс отложено количество процессорных модулей, измеряемое в штуках, по оси ординат – время выполнения алгоритма размещения, выраженное в тактах. Каждый столбец гистограммы показывает объём данных, при которых проводилось моделирование. Синий – 100 Мб, оранжевый – 100000 Мб, серый – 10000000 Мб. По общей динамике видно, что с увеличением процессорных модулей, время выполнения алгоритма растёт. Если рассмотреть метод ветвей и границ, результаты которого представлены на рисунке 4, видно, что время выполнения этого метода измеряется порядком 10^9 степени тактов, в то время как разработанный в данной работе алгоритм работает намного быстрее – за 10^4 тактов. Значение времени растёт линейно, а не экспоненциально.

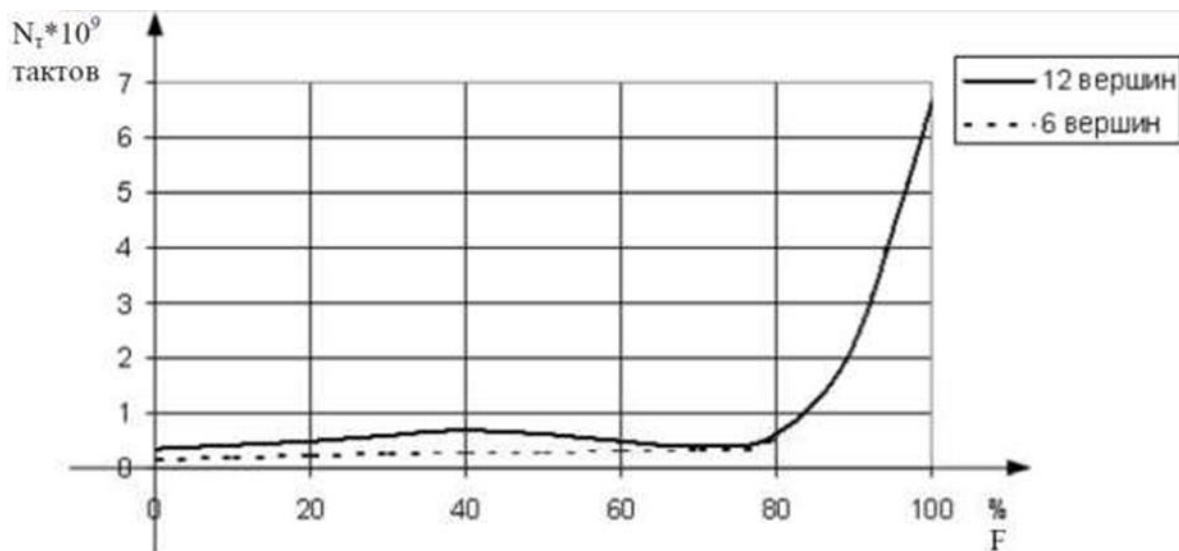


Рисунок 4 – Результаты размещения метода ветвей и границ

На рисунке 4 по оси ОХ отложено относительное количество дуг F от максимально возможного числа дуг в графе G , выраженное в процентах. По оси ординат отложено количество тактов работы процессора N_T , требующихся для решения задачи планирования размещения.

Для получения результатов был проведен ряд испытаний. Менялось число процессорных модулей и подсчитывалось, сколько связанных пар задач было при размещении. Так же фиксировался выигрыш. Результаты данного исследования представлены на рисунке 5.

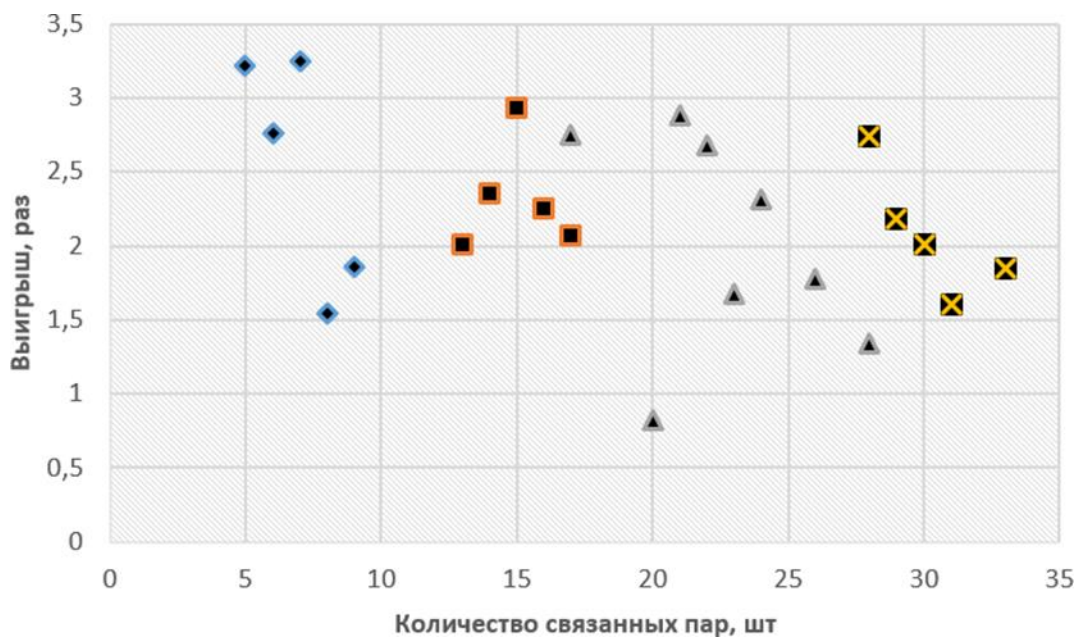


Рисунок 5 – График зависимость выигрыша от числа связанных пар

На графике, представленном на рисунке 5 по оси абсцисс отложено количество связанных пар задач, по оси ординат – выигрыш. Геометрическими фигурами на графике показаны значения, относящиеся к определенному числу программных модулей: круги – 25 процессорных модулей, квадраты – 50 процессорных модулей, треугольники – 75 процессорных модулей и крестики – 100 процессорных модулей.

По графику можно заметить общую закономерность: с увеличением количества связанных задач, выигрыш уменьшается. Это говорит о том, что распределять в динамической системе эффективнее не связанные задачи.

Заключение

По результатам моделирования были сделаны следующие выводы:

1. С увеличением количества процессорных модулей коэффициент, показывающий во сколько раз уменьшилось время передачи данных после размещения, уменьшается линейно, но не достигает нулевого значения. Скорость

выполнения разработанного алгоритма примерно в 100 раз выше алгоритмов размещения, основанных на методе ветвей и границ.

2. С увеличением объема передаваемых данных, выигрыш растёт до значения 2, а затем колеблется в пределах $2 \pm 0,5$. Наибольший выигрыш (2,5-3 раза) достигается при показателях: объем данных задач равен 10000 Мб и количество процессорных модулей равно 75-100.

3. С увеличением количества связанных пар задач линейно уменьшается выигрыш.

Библиографический список

1. Андреев А.М., Можаров Г.П., Сюзев В.В. Многопроцессорные вычислительные системы: теоретический анализ, математические модели и применение. - М.: Изд-во МГТУ им. Н. Э. Баумана, 2011. - 332 с.
2. Кондрашин М.А., Арсенов О.Ю., Козлов И.В. Применение технологии виртуализации и облачных вычислений при построении сложных распределенных моделирующих систем // Труды МАИ. 2016. № 89. URL <http://trudymai.ru/published.php?ID=73411>
3. Богданов А.В., Корхов В.В., Мареев В.В. Архитектуры и топологии многопроцессорных вычислительных систем. - М.: Изд-во НОУ ИНТУИТ, 2016. - 136 с.
4. Борзов Д.Б. Аппаратные средства планирования размещения задач в мультипроцессорных системах критического назначения: монография. – Курск: Юго-Западный государственный университет, 2018. - 179 с.

5. Борзов Д.Б., Титов В.С. Вопросы проектирования и динамической реконфигурации топологии систем логического управления в системах высокой готовности: монография. – Курск: Юго-Западный государственный университет, – 282 с.
6. Борзов Д.Б., Титов В.С. Параллельные вычислительные системы (архитектура, принципы размещения задач. – М.: Изд-во LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. - 152 с.
7. Власов Д.В. Современные проблемы информатики: философский анализ // Статистика и экономика. 2011. № 2. С. 233 – 239.
8. Гришанков В. Технология ANT – что это такое в телефоне? URL: <https://androidlime.ru/ant-technology-what-is-it-on-the-phone>
9. Евдокименков В.Н., Ляпин Н.А. Минимаксная оптимизация маневров преследования противника в условиях ближнего воздушного боя // Труды МАИ. 2019. № 106. URL: <http://trudymai.ru/published.php?ID=105735>
10. Осипов Н.А., Шавин А.С., Тарасов А.Г. Методика идентификации помех, действующих в канале передачи информации робототехнических систем // Труды МАИ. 2017. № 94. URL: <http://trudymai.ru/published.php?ID=81085>
11. Закалюжный А.А. Развитие беспроводных сетей как средство контроля и управления удаленными системами // Молодой исследователь Дона. 2018. № 4 (13). С. 46 - 51.
12. Зинкин С.А., Белецкий П.А. Оптимизация размещения данных по узлам информационно-вычислительной сети // Материалы II Международной научной

- конференции «Современные тенденции технических наук» (Уфа, 20-23 мая 2013). – Уфа: Изд-во Лето, 2013. С. 29 – 31.
13. Князева М.В. Метод ветвей и границ для решения задачи сетевого планирования с ограниченными ресурсами // Известия ЮФУ. Технические науки. 2010. № 7 (108). С. 78 – 84.
14. Липницкий В.А., Сергей А.И., Спичекова Н.В. Научно-технические задачи, связанные с бинарными матрицами // Материалы научно-технического семинара «Цифровая обработка сигналов и теория кодирования» (Минск, 26 апреля 2018). – Минск: Белорусский государственный университет информатики и радиоэлектроники, 2018. С. 43 - 47.
15. Наумов А.В. Сай Кхин Аунг Тинт. Об адаптации обучающих и тестирующих систем переподготовки молодых специалистов на предприятиях аэрокосмического комплекса // Труды МАИ. 2011. № 42. URL: <http://trudymai.ru/published.php?ID=24321>
16. Новиков Е.А. Ващенко Г.В. Последовательный и параллельный алгоритмы переменного шага на основе (2,2)-метода // Современные проблемы науки и образования. 2011. № 5. URL: <http://www.science-education.ru/ru/article/view?id=4814>
17. Семахин А.М., Баталов И.С. Динамическое программирование в решении задачи оптимального размещения электронных компонентов системы управления // Молодой ученый. 2013. № 6. С. 144 - 146. URL: <https://moluch.ru/archive/53/7243/>
18. Чистобаев А.И., Семенова З.А. Статистический метод в медико-географических исследованиях // Географический вестник. 2013. № 1 (24). С. 18 – 25.
19. Шейкин М. Сетевые технологии ZigBee. Обзор элементной базы // Телекоммуникации и связь. 2011. № 6. С. 36 – 40.

20. Yanovskii A.A., Simonovskii A.Ya., Klimenko E.M. On the Influence of the Magnetic Field upon Hydrogasdynamic Processes in a Boiling Magnetic Fluid // Surface Engineering and Applied Electrochemistry, 2014, vol. 50, no. 3, pp. 260 - 266.

Modeling a reconfigurable real-time system with a decrease in the time spent on processing information

Borzov D.B.^{1*}, Koshelev M.A.^{1}, Sokolova Yu.V.^{2***}**

¹*South-Western State University, 94, 50-let Oktyabrya str., Kursk, 305040, Russia*

²*Lavochkin Research and Production Association, NPO Lavochkin, 24, Leningradskay str., Khimki, Moscow region, 141400, Russia*

**e-mail: borzovdb@kursknet.ru*

***e-mail: maks46.krutoe@gmail.com*

****e-mail: jv.sokolova@mail.ru*

Abstract

The purpose of the research is to reduce the time spent on processing information, contributing to an increase in the speed of a reconfigurable real-time system, by creating a methodology and an algorithm for distributing an array of computational tasks.

The research methods of the work are based on the definitions of set theory, graphs, probability theory and mathematical statistics. With their help, a mathematical model and a system of criteria for a reconfigurable real-time computing system, built on a wireless protocol, was created, which allows placing tasks according to the minimum-maximum time estimate. And also an algorithm for placing tasks in a computer complex is proposed, which allows increasing the performance of the system by reducing the time of data transfer.

As a result of the research, a method was developed for the distribution of computational tasks in a real-time reconfigurable computing system, which makes it possible to reduce the time of data transmission within the system, which increases the performance of the complex. A methodology for distributing computational tasks to individual processor modules has been compiled. The work of the proposed methods is

simulated. And a system of criteria is proposed that describes the final state of a reconfigurable real-time computing system. As a result, the resulting system gave a gain in time of about 2.5 times. Placing connected problems in dynamic systems reduces the payoff coefficient linearly.

The resulting model can be used in distributed computing. The work of the proposed methods is simulated. Their analysis confirmed the increase in system performance as a whole.

Keywords: algorithm, wireless protocol, multiprocessor system, reconfigurable computing system.

References

1. Andreev A.M., Mozharov G.P., Syuzev. V.V. *Mnogoprotsessornye vychislitel'nye sistemy: teoreticheskii analiz, matematicheskie modeli i primeneniye* (Multiprocessor computing systems: theoretical analysis, mathematical models and application), Moscow, Izd-vo MGTU im. N. E. Baumana, 2011, 332 p.
2. Kondrashin M.A., Arsenov O.Yu., Kozlov I.V. *Trudy MAI*, 2016, no. 89. URL <http://trudymai.ru/eng/published.php?ID=73411>
3. Bogdanov A.V., Korkhov V.V., Mareev V.V. *Arkhitektury i topologii mnogoprotsessornykh vychislitel'nykh system* (Architecture and topology of multiprocessor computing systems), Moscow, Izd-vo NOU INTUIT, 2016, 136 p.
4. Borzov D.B. *Apparatnye sredstva planirovaniya razmeshcheniya zadach v mul'tiprotsessornykh sistemakh kriticheskogo naznacheniya* (Hardware planning of tasks

placement in multiprocessor systems for critical purposes), Kursk, Yugo-Zapadnyi gosudarstvennyi universitet, 2018, 179 p.

5. Borzov D.B., Titov V.S. *Voprosy proektirovaniya i dinamicheskoi rekonfiguratsii topologii sistem logicheskogo upravleniya v sistemakh vysokoi gotovnosti* (Issues of design and dynamic reconfiguration of the topology of logical control systems in high availability systems) Kursk, Yugo-Zapadnyi gosudarstvennyi universitet, 282 p.

6. Borzov D.B., Titov V.S. *Parallel'nye vychislitel'nye sistemy (arkhitektura, printsipy razmeshcheniya zadach* (Parallel computing systems (architecture, principles of task placement), Izd-vo LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012, 152 p.

7. Vlasov D.V. *Statistika i ekonomika*, 2011, no. 2, pp. 233 – 239.

8. Grishankov V. *Tekhnologiya ANT – chto eto takoe v telefone?* URL: <https://androidlime.ru/ant-technology-what-is-it-on-the-phone>

9. Evdokimenkov V.N., Lyapin N.A. *Trudy MAI*, 2019, no. 106. URL: <http://trudymai.ru/eng/published.php?ID=105735>

10. Osipov N.A., Shavin A.S., Tarasov A.G. *Trudy MAI*, 2017, no. 94. URL: <http://trudymai.ru/eng/published.php?ID=81085>

11. Zakalyuzhnyi A.A. *Molodoi issledovatel' Dona*, 2018, no. 4 (13), pp. 46 - 51.

12. Zinkin S.A., Beletskii P.A. *Materialy II Mezhdunarodnoi nauchnoi konferentsii «Sovremennye tendentsii tekhnicheskikh nauk»*, Ufa, Izd-vo Leto, 2013, pp. 29 – 31.

13. Knyazeva M.V. *Izvestiya YuFU. Tekhnicheskie nauki*, 2010, № 7 (108), pp. 78 – 84.

14. Lipnitskii V.A., Sergei A.I., Spichkova N.V. *Materialy nauchno-tekhnicheskogo seminara «Tsifrovaya obrabotka signalov i teoriya kodirovaniya»*, Minsk, Belorusskii gosudarstvennyi universitet informatiki i radioelektroniki, 2018, pp. 43 - 47.

15. Naumov A.V. Sai Kkhin Aung Tint. *Trudy MAI*, 2011, no. 42. URL: <http://trudymai.ru/eng/published.php?ID=24321>
16. Novikov E.A. Vashchenko G.V. *Sovremennye problemy nauki i obrazovaniya*, 2011, no. 5. URL: <http://www.science-education.ru/ru/article/view?id=4814>
17. Semakhin A.M., Batalov I.S. *Molodoi uchenyi*, 2013, no. 6, pp. 144 - 146. URL: <https://moluch.ru/archive/53/7243/>
18. Chistobaev A.I., Semenova Z.A. *Geograficheskii vestnik*, 2013, no. 1 (24), pp. 18 – 25.
19. Sheikin M. *Telekommunikatsii i svyaz'*, 2011, no. 6, pp. 36 – 40.
20. Yanovskii A.A., Simonovskii A.Ya., Klimenko E.M. On the Influence of the Magnetic Field upon Hydrogasdynamic Processes in a Boiling Magnetic Fluid, *Surface Engineering and Applied Electrochemistry*, 2014, vol. 50, no. 3, pp. 260 - 266.