

Метод и методика беспроводной передачи данных в мультипроцессорных системах для нестационарных объектов обмена

Борзов Д.Б.^{1*}, Дюбрюкс С.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: serhio5551@yandex.ru*

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

Статья поступила 02.09.2020

Аннотация

Статья посвящена нестационарным мультипроцессорным системам. Затрагивается вопрос о беспроводной передаче данных между процессорными объектами (ПО), такими как ноутбуки, нетбуки, тачпады, планшеты, мобильные устройства и т.п., способными передавать информацию на расстоянии по беспроводной среде передачи данных. Например, такой сетью может быть среда передачи данных Wi-Fi. При этом сами ПО могут независимо перемещаться в пространстве. В данном случае важна возможность изменения расстояния между парами ПО, что влияет на объемы передаваемой информации, их скорости и, как следствие, общее время передачи в целом.

Ключевые слова: беспроводная сеть, Wi-Fi, процессорные объекты, алгоритм обмена информацией, метод и методика беспроводной передачи данных, нестационарные объекты обмена, размещение задач.

Постановка задачи

Разрабатывается метод и методики беспроводной передачи данных с возможностью учета перемещения в пространстве объектов обмена, обеспечивающего повышение производительности мультипроцессорных систем и снижение общей величины коммуникационной задержки.

С учетом специфики решаемой в работе задачи необходимо выполнять размещение задач (программ, подпрограмм, файлов и т.п.) между множеством ПО мультипроцессорной системы с учетом текущего расстояния между ПО таким образом, чтобы снизить суммарную коммуникационную задержку, повысить производительность и увеличить скорость выполнения всей задачи в целом.

В статье авторами предлагается метод и методика беспроводной передачи данных в мультипроцессорных системах для перемещаемых объектов обмена. Предложенные идеи предполагается использовать в мультипроцессорных системах критического назначения (системы слежения, наблюдения, прицеливания, контроля и т.д.), когда время на внештатную ситуацию в системе должно быть минимальным. Средой передачи данных для ПО предполагается беспроводная сеть Wi-Fi, что особенно важно, например, для бортовых систем, что одновременно накладывает ограничения на скорости и объемы передаваемых данных, вызванного спецификой беспроводного протокола Wi-Fi.

Введение

При построении вычислительных систем (ВС) возникает проблема совместимости ее архитектуры со структурой обрабатываемых задач [1, 2]. Это

проявляется в том, что любая ВС выполняет поставленную задачу только для определенного класса алгоритмов и методов. В случае, если структура задания и архитектура системы не соответствуют друг другу, производительность всей ВС минимальна [3].

Так как не существует структуры, одинаково хорошо обрабатывающей задания разных типов, при построении вычислительных систем используют несколько видов топологий, соотносящихся с определенным классом алгоритмов, поэтому в рамках данной работы необходимо рассмотреть, как топологии ВС, так и методы размещения, чтобы добиться максимальной производительности [4]. Необходимо учитывать, что ВС строится на беспроводной технологии [5-7]. Следовательно, необходимо провести анализ протоколов подобных способов связи и выявить совместимость их способов соединения с топологиями ВС.

Мультипроцессорная система, состоящая из множества ПО, представляется неориентированным графом $G=(V,R)$, где V – множество ПО, R – ребра графа [8].

Пример графа системы представлен на рисунке 1.

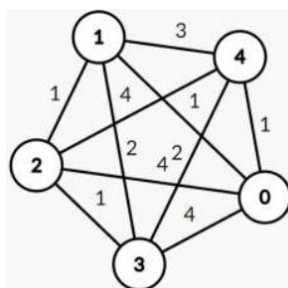


Рис. 1. Пример графа системы

На рисунке 1 круги обозначают ПО, числа в них означают идентификатор ПО, ребра графа показывают связи между ПО, а цифры над дугами показывают эквиваленты расстояний между ПО.

Например, система, показанная на рисунке 1, определяется множеством $v = \{1,2,3,4,5\}$, а множество $R = \{1,4,3,1,1,2,3,1,4,2\}$.

В стандарте IEEE 802.11ac [7-9] применяется таблица коэффициентов MCS, представленная в таблице 1.

Таблица – Индексы MCS

MCS	4		3		2		1	
	20 МГц		40 МГц		80 МГц		160 МГц	
	Скорость	RSSI, дБ	Скорость	RSSI, дБ	Скорость	RSSI, дБ	Скорость	RSSI, дБ
0	13	-82	27	-79	58.5	-76	117	-73
1	26	-79	54	-76	117	-73	234	-70
2	39	-77	81	-74	175.5	-71	351	-68
3	52	-74	108	-71	234	-68	468	-65
4	78	-70	162	-67	351	-64	702	-61
5	104	-66	216	-63	468	-60	936	-57
6	117	-65	243	-62	526.5	-59	1053	-56
7	130	-64	270	-61	585	-58	1170	-55
8	156	-59	324	-56	702	-53	1404	-50
9			360	-54	780	-51	1560	-48

В таблице 1 первый столбец обозначает индексы MCS, первая строка, присвоенные частотным каналам номера, вторая строка – частота каналов, выраженная в МГц, третья строка – скорость передачи данных в Мб/с и сила сигнал в дБ. На пересечении строк со столбцами представлены значения скорости передачи данных и силы сигнала MCS.

Из таблицы 1 видна сила сигнала и канал, на котором работает ПО, который может указать на предпочтительный ПО, используя показатель коэффициент расстояния (КР):

$$KP = -F \cdot RSSI, \quad (1)$$

где F – номер, присвоенный каналу, $RSSI$ – значение силы сигнала.

Тогда применяя (1) к показателям графа, представленного на рисунке 1, получаем результаты, представленные в таблице 2.

Таблица 2 – Коэффициенты расстояния

MCS	4		3		2		1	
	20 МГц		40 МГц		80 МГц		160 МГц	
	Скорость	RSSI, ДБ	Скорость	RSSI, ДБ	Скорость	RSSI, ДБ	Скорость	RSSI, ДБ
0	13	-82	27	-79	58.5	-76	117	-73
1	26	-79	54	-76	117	-73	234	-70
2	39	-77	81	-74	175.5	-71	351	-68
3	52	-74	108	-71	234	-68	468	-65
4	78	-70	162	-67	351	-64	702	-61
5	104	-66	216	-63	468	-60	936	-57
6	117	-65	243	-62	526.5	-59	1053	-56
7	130	-64	270	-61	585	-58	1170	-55
8	156	-59	324	-56	702	-53	1404	-50
9			360	-54	780	-51	1560	-48

В таблице 2 – первый столбец обозначает индексы MCS, а в остальных столбцах показаны частотные номера каналов. На пересечении строк и столбцов показаны рассчитанные коэффициенты расстояния.

Таким образом из таблицы можно оценить расстояния до любого устройства в мультипроцессорной системе. В результате получаем граф, изображенный на рисунке 2.

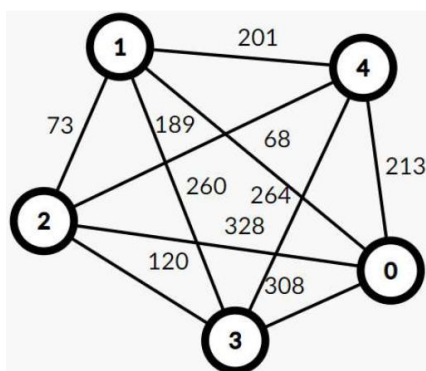


Рис. 2. Расстояния между процессорами с учетом (1)

На рисунке 2 в кругах показаны процессорные модули, числа в них – идентификатор ПМ, ребра показывают связи между устройствами, а цифры над дугами графа означают эквивалент расстояния.

Таким образом из анализа рисунка 2 получаем множество $V = \{1,2,3,4\}$ и множество $R = \{68,328,308,213,73,260,201,120,189,264\}$. Тогда по данным из V и R получаем матрицу смежности $A_{5 \times 5}$ [14], где элементы a_{ij} – значения множества R :

$$A = \begin{pmatrix} 0 & 63 & 328 & 308 & 213 \\ 63 & 0 & 73 & 260 & 201 \\ 328 & 73 & 0 & 120 & 189 \\ 308 & 260 & 120 & 0 & 264 \\ 213 & 201 & 189 & 264 & 0 \end{pmatrix} \quad (2)$$

Назовем матрицу (2) – матрицей расстояний (MP).

Среди множества процессорных модулей выбираем процессорный модуль – мастер. В его функции входит:

- отслеживание изменений в архитектуре ВС;
- распределение вычислительных задач для уменьшения временных затрат;
- опрос внешнего источника (сервера, суперкомпьютера, кластера, базы данных и т.п.) на наличие заданий;

– распределение полученных задач на процессорные модули, включенные в систему в момент времени t .

Задания приходят мастеру в виде массива:

$$Task[N] = \{\{p_k; v_k\} \dots \{p_k; v_k\}\}, \quad (3)$$

где p_k – связь с другой задачи из массива, k – номер задачи равный $\overline{1, n}$, v_k – объем данных задач.

Например, $Task[5] = \{\{1;10\}, \{1;15\}, \{3;10\}, \{4;8\}, \{3;5\}\}$. По данному массиву известно, что задачи 1-2 и 3-5 связаны друг с другом, а задача 4 не зависит от других элементов массива.

Массив заданий представляется множествами:

- P мощностью $|P| = N$;

- V мощностью $|V| = N$.

По $Task[5]$ получаем:

- $P = \{1, 1, 3, 4, 3\}$,

- $V = \{10, 15, 10, 8, 5\}$.

Множества запишем в виде V – матрицы объемов заданий (МОЗ) и P – матрицы связей (МС):

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, V = \begin{pmatrix} 10 & 15 & 10 & 8 & 5 \\ 15 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 1 \\ 8 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (4)$$

МОЗ показывает какой из программных модулей является мастером, что определяется по ненулевому столбцу или строке. В примере (см. формулу 4) мастер – первый ПМ. Так же матрица V показывает первоначальное размещение ПО. MC показывает связь между задачами. Если $v_{ij}=1$, то задача с порядковым номером i связана с задачей под номером j .

Так как мультипроцессорная система динамическая, то для работы с ней необходимо описать ее конечное состояние и систему критериев.

Пусть конечное состояние системы – это состояние, при котором мастер начинает распределять задачи, обозначенные как $S_t(N)$, где N – количество ПО, t – время ожидания построения системы. Так t варьируется соотношением $[t_{\min}; t_{\max}]$, где t_{\min} – минимальное время ожидания, t_{\max} – максимальное время ожидания.

ПМ соединяются средствами протокола Wi-Fi [9–11]. В этом случае на одну точку доступа теоретически можно подключить 254 устройства, но, так как предполагается, что все устройства работают в режиме Ad-Нос, описанном в первом разделе, количество устройств не устанавливается, а варьируется от 3 до 254. Так как конечное состояние системы – это набор ее свойств, при котором можно начинать размещение, записывается оно следующей схемой:

$$S_t(N) = \begin{cases} t = t_{\text{ук}}, t_{\min} < t_{\text{ук}} < t_{\max} \\ N = N_p, 3 < N_p < 254 \\ A_{N \times N}, a_{ij} \in KP \end{cases} \quad (5)$$

Для минимизации времени выполнения задания необходимо сократить время пересылки данных от мастера к ПО, ответственному за выполнение. Скорость передачи данных C зависит от расстояния между источником и приемником:

$$KP \rightarrow 0 \Rightarrow C \rightarrow C_{\text{раб}} \quad (6)$$

Время передачи данных рассчитывается по формуле:

$$t_{\text{передачи}} = \frac{V}{C} \quad (7)$$

где V – объем передаваемых данных, C – скорость передачи данных.

Тогда из (7) следует $C \rightarrow C_{\text{раб}}$, $V \rightarrow V_{\text{min}}$. Следовательно, $t \rightarrow t_{\text{передачи}}$.

Так как объем данных задачи $\{v_{ij}\}$ – величина постоянная, то, очевидно, что уменьшение времени передачи обеспечивается передачей мастером самых объемных задач процессорным модулям, у которого KP в матрице расстояний наименьший, что говорит о наибольшей скорости передачи данных. Так, например, если рассмотреть матрицу из формулы 2 и предположить, что мастером является 1 процессорный модуль, то задачу с весом 15 он назначает себе, а задачу с весом 10 можно распределить на 0 и 3 ПО. Таким образом, при распределении задач мастеру необходимо учитывать объемы данных, KP ПМ и связность задач.

Учитывая вышесказанное, составляющие задачи оптимального размещения можно сформулировать как [12]:

$$K_y = \max \left(\frac{m_{ij}}{c} \right) \quad (8)$$

где m_{ij} – объем передаваемой задачи ($i, j = \overline{0, N}$), N – число процессорных модулей, C – скорость передачи данных между мастером и процессорным модулем, которому назначена задача объемом m_{ij} .

По завершению задачи ПО либо отправит данные инициатору (серверу, базе данных, кластеру и т.п.) задания, либо передаст данные другому ПО, которому для выполнения полученного задания требуется результат выполнения текущей задачи.

Результаты и их обсуждение

При построении реконфигурируемой вычислительной системы критического назначения необходимо выполнить действия, направленные на составление матрицы расстояний $A_{N \times N}$ и выбор мастера. MP содержит информацию о расстоянии между процессорными модулями, мастер же отвечает за запрос и размещение задач процессорным модулям, входящим в систему на текущий момент времени.

При этом мастер выбирается из условий:

1) $r = \min \left\{ \sum_{i,j=0}^N r_{i,j} \right\}$, где N – количество ПМ, r_{ij} – значение MP .

2) $r_n = \text{count}(r_{\text{max}})$ – максимум в строке r при выполнении условия $r_M < r_n$.

3) $G_M > G_{\text{ПМ}}$, где G_M – коэффициент усиления антенны мастера, $G_{\text{ПМ}}$ – коэффициент усиления антенны процессорного модуля, претендующего на роль мастера.

Здесь первое условие является необходимым и достаточным, а второе и третье условие используются как дополнительные, когда первое условие не выявило мастера.

По приведенным условиям алгоритм выбора мастера из множества устройств состоит из следующих шагов [13]:

1. По каждой строке матрицы расстояний (МР) считается сумма и высчитывается максимум во всей МР.

2. Из полученных сумм выбирается минимальная. Если минимальных сумм несколько, то присутствует несколько претендентов на роль мастера, и в этом случае проверяется следующее условие.

3. Для устройств, где сумма по строке в МР минимальна, считается количество максимумов, найденных на первом шаге, после чего данный показатель сравнивается. При этом наименьший максимум показывает мастера. Если же и это не выявило мастера, то выполняется четвертый шаг.

4. Для устройств, у которых минимум максимумов в строке, сравнивается коэффициент усиления антенны. Устройство с самым большим коэффициентом усиления показывает мастера.

Для составления матрицы расстояний необходимо знать уровень сигнала. Этот показатель, наравне с физическим расстоянием, влияет на скорость передачи данных, и вычисляется при подключении устройства к точке доступа. В свою очередь, расчет расстояния между устройствами является трудоемким процессом и дает результат с точностью в пределах десяти метров. Поэтому в МР будет записываться КР, как было показано выше, а не физическое расстояние.

Таким образом, методика построения реконфигурируемой критической вычислительной системы состоит из следующих шагов [11-15]:

1. За время ожидания в системе фиксируется количество активных устройств.
2. Составляется матрица расстояний из КР.
3. Выбирается мастер.

Так как процессорные модули работают в режиме точки доступа, то каждый ПМ при активном Wi-Fi начинает строить РВСРВ. Поэтому на первом шаге каждому устройству, подключающемуся к системе, присваивается идентификатор, уникальный в пределах действия мультипроцессорной системы.

На протяжении времени ожидания процессорные модули обмениваются служебными пакетами, содержащими идентификаторы ПМ, коэффициент усиления антенны и массив информации. Пакет имеет вид, представленный на рисунке 3.

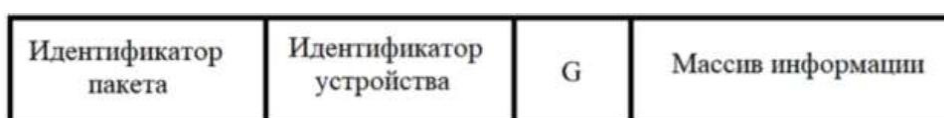


Рис. 3. Служебный пакет обмена

На рисунке 3 идентификатор пакета – принимает значение `exchang`, идентификатор устройства – номер устройства, присвоенный при появлении ПМ в системе.

Массив информации имеет вид:

$$dev[N] = \{ \{id_i; sig_i\} \dots \{id_i; sig_i\} \}, \quad (9)$$

где `id` – идентификатор процессорного ПО, `sig` – коэффициент расстояния, `G` – коэффициент усиления антенны.

Например:

<code>exchang</code>	1	<code>{{1;308},{2;256},{4;48}}</code>	1,5
----------------------	---	---------------------------------------	-----

На шаге 2 после обмена пакетами у каждого ПМ сформируется матрица расстояний:

$$A = \begin{pmatrix} 0 & 308 & 183 & 128 \\ 308 & 0 & 256 & 48 \\ 183 & 256 & 0 & 142 \\ 128 & 48 & 142 & 0 \end{pmatrix} \quad (10)$$

Кроме того, сформируется таблица:

Таблица 3. Коэффициент усиления антенн

id	G
0	1
1	2
2	1.5
3	3

В таблице 3 первый столбец – это идентификатор устройства id, а второй – это коэффициент усиления антенны G.

На шаге три считается сумма элементов каждой строки матрицы A. По (8) и матрице A получаем:

$$r_{sum_0} = 491, r_{sum_1} = 612, r_{sum_2} = 581, r_{sum_3} = 318.$$

По условию 1 мастером становится процессорный модуль 3.

После выполнения процедуры выбора процессора-мастера, начинается размещение полученных заданий на ПМ мультипроцессорной системы.

Для выполнения этой функции предложена 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 [11-20].

Нижняя оценка времени – недостижимое значение в силу того, что рассылка заданий производится мастером, и расстояния до каждого ПМ фиксированы. При расчете K_{y_min} выбираются минимальные значения расстояний из всего множества

значений. Таким образом, поиск нижней оценки времени передачи сводится к выполнению следующих шагов:

1. Отсортировать объемы заданий по убыванию.
2. Записать в \bar{A} значения коэффициентов расстояний в порядке возрастания.
3. Исключить из \bar{A} все нули.
4. Перевести \bar{A} [0] в Мб/с.
5. Найти K_{y_min} как частное максимального объема данных на полученное значение скорости передачи данных.

Полученное значение является минимальным временем передачи данных.

Для выполнения размещения, необходимо учитывать расстояния, доступные мастеру и возможные цепочки пересылок через другие ПМ, способные сократить время. 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:

$$K_{y_T} = \max \left(\frac{v_{ij}}{a_{ij}} \right) \quad (11)$$

где a_{ij} – переведенный 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

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

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

$\forall a_{ij}$ в $\vec{A} = \|v_{ij}\|$ так, что $a_{ij}^{z_1} \geq a_{ij}^{z_2} \Leftrightarrow z_1 < z_2$, 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 А: $C = \vec{A}[0]$.

5. Перевести С в Мб/с, используя таблицы 1-2.

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

Этап 3: Выполняется размещение задач так что $v_{z1} \geq v_{z2}$ и $a_{z1} \geq a_{z2}$

Этап 4: Вычисляется время пересылки:

$$K_{y_H} = \max\left(\frac{v_{mj}}{a_{mj}}\right) + \frac{v_{ij}}{a_{ij}} \quad (12)$$

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

Этап 5: Анализ величины η по формулам:

$$\eta_{H_min} = \frac{k_{y_H}}{k_{y_min}} \quad (13)$$

$$\eta_{T_min} = \frac{k_{y_T}}{k_{y_min}} \quad (14)$$

$$\eta_w = \frac{\eta_{T_min}}{\eta_{H_min}} \quad (15)$$

Коэффициент приближения η_{H_min} показывает, во сколько раз размещение приблизило время пересылки к минимуму, коэффициент начального размещения η_{T_min} означает, во сколько раз начальное размещение хуже минимального, а выигрыш η_w – мин-максный критерий, указывающий во сколько раз новое размещение выгоднее начального.

Было проведено программное моделирование предложенного метода и подхода, в результате чего были получены следующие зависимости: выигрыша от количества процессоров и выигрыша от максимального объема передаваемых данных. Кроме того, был проведен анализ скорости выполнения алгоритма, выраженная в тактах относительно количества процессорных модулей [21-23].

Была получена зависимость роста полученного выигрыша от количества ПМ, представленная на рисунке 4.

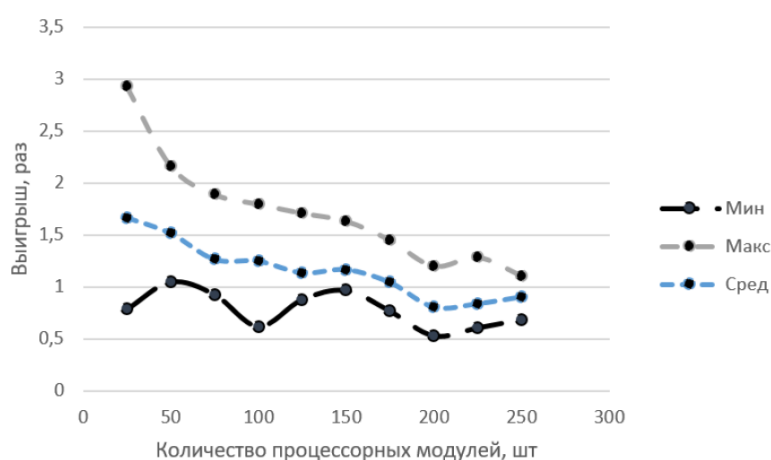


Рис. 4. Рост полученного выигрыша от количества ПМ

На рисунке 4 получаемый выигрыш разделен на три группы: минимальный, максимальный и средний. Из анализа зависимостей можно сделать вывод, что все три показателя стремятся к единице, но дальнейшее увеличение количества процессорных модулей не уменьшает выигрыш до нулевого значения.

Проведено моделирование с изменением объема передаваемых данных от 100 до 100000000 Мб с количеством ПМ от 25 до 250. В результате получена зависимость выигрыша от объема передаваемых данных, показанная на рисунке 5.

	25	50	75	100	125	150	175	200	225	250
100	0.894	0.979	1.436	1.005	1.221	1.051	0.779777	0.888198	1.25331	0.729
1000	2.284	1.705	1.413	1.457	1.620	1.876	1.94515	1.21942	1.46048	1.565
10000	1.183	1.803	2.768	2.093	2.040	2.018	2.36914	1.85957	1.44741	2.018
100000	0.686	1.468	1.936	2.408	1.958	1.812	1.65331	1.98876	1.68993	1.765
1000000	2.513	1.687	2.293	2.751	2.130	1.465	1.79025	1.51268	1.80964	1.437
10000000	1.289	1.530	2.552	1.405	1.543	1.981	1.5464	1.53473	1.5335	1.598
100000000	2.581	1.671	1.936	1.538	1.837	1.693	1.52022	1.59578	1.55986	1.785

Рис. 5. Зависимость выигрыша от объема передаваемых данных

На данном рисунке первый столбец – объемы задач, указанного в мегабитах, в строках отложено количество процессорных модулей. На пересечении указаны значения выигрыша. Красным отмечена область, где коэффициент меньше значения 1,7. Выше этого значения отмечено зеленым. По таблице значений можно заметить, что наиболее выгодными являются следующие показатели: количество процессорных модулей от 50 до 150 и объем заданий от 10000 до 1000000 Мб. Это объясняется особенностями беспроводного соединения. Кроме того, с ростом объема данных растет и выигрыш, пока объем данных не достигает значения 10000 Мб. Далее коэффициент колеблется на $\pm 0,5$, но и не уменьшается, и не увеличивается больше этого значения, что говорит о том, что показатель установился на отметке $2 \pm 0,5$ раза.

Кроме того, промоделирована передача данных с объемами 10000 и 1000000 Мб. При этих показателях получен самый высокий минимум и максимум выигрыша. Вследствие этого было промоделировано увеличение количества процессорных модулей, при фиксированном объеме данных и получены результаты, представленные на рисунках 6 и 7.

	25	50	75	100	125	150	175	200	225	250
1	0.937817	0.866891	1.98183	2.77975	1.76727	2.02454	1.97807	2.19356	1.78882	2.11016
2	1.42917	2.45668	1.69201	1.68495	2.65131	2.78234	1.91135	1.96132	1.88089	1.90367
3	1.77561	1.64904	2.64069	3.05383	1.77334	2.01431	2.06632	1.7323	1.87245	1.99803
4	2.26953	3.08685	1.71496	2.70207	3.03609	2.08347	2.31534	1.82739	2.30004	1.32628
5	1.61555	2.4182	2.18639	2.27593	1.9153	1.34388	2.16559	2.30113	2.49682	1.55565
6	1.18977	2.15685	2.23294	2.03341	2.2205	1.62616	1.83855	2.55014	1.90958	1.64652
7	2.05179	2.76859	1.54856	2.10501	1.61811	1.24003	1.81074	1.73012	1.53201	1.76287
8	2.03984	2.79462	2.16029	2.5102	2.62739	2.11183	2.32018	2.05798	1.68825	1.91391
9	2.08835	0.979372	1.84298	2.78701	2.12069	2.65242	2.0493	1.63436	2.14108	1.74339
10	2.70812	2.25565	1.45229	2.48466	1.81274	2.02362	1.72702	2.26868	1.93873	2.28424

Рис. 6. Результаты моделирования при объеме данных 10000 Мб

	25	50	75	100	125	150	175	200	225	250
1	2.07377	2.2574	2.53749	2.12369	1.65274	1.99058	1.11013	1.75453	1.78774	1.70861
2	0.895322	3.15775	1.09957	1.57468	1.7982	1.67548	1.58767	1.75787	1.8349	1.88168
3	2.26321	1.98203	0.854117	1.99171	1.58679	1.25769	2.04884	1.49514	1.99865	1.39076
4	1.78347	3.20989	1.70289	1.89388	1.97545	1.57572	1.5866	2.01834	1.33286	1.50766
5	2.94601	2.39449	1.74782	2.00592	2.5097	1.41604	2.27802	1.72519	1.76049	1.4848
6	1.08923	1.65832	1.60968	2.07393	1.33746	1.72714	1.45337	1.4507	1.72942	1.48613
7	1.4228	2.00123	1.32889	1.39442	1.76985	2.12062	2.62012	1.84726	1.59698	1.81034
8	1.24052	1.58833	2.1153	1.7568	2.35088	1.79359	1.28211	2.22382	1.85845	1.68446
9	1.75917	1.80772	2.5891	2.19784	2.44385	2.17452	1.14355	1.76966	1.53229	1.42863
10	1.73647	1.32338	2.12798	1.73896	1.22668	2.11163	1.54558	1.70433	1.58355	1.44935

Рис. 7. Результаты моделирования при объеме данных 1000000Мб

На рисунках 6-7 первый столбец – номер испытания, в строках показано количество процессорных модулей. На пересечении – значения выигрыша. Зеленым цветом отмечены значения, которые превышают значение, равное 1,7. Уже в первом десятке испытания виден результат: задачи с объемом данных равным 1000 Мб распределяются значительно лучше – выигрыш составляет 2-3 раза. При этом количество процессорных модулей может варьироваться от 100 до 200. А задачи с объемом 1000000 Мб распределяются хуже – выигрыш в пределах 1-1,7 раза. При этом с таким объемом данных лучше всего справляется система из 50 процессорных модулей.

Из рисунков 6 и 7 следует вывод, что наиболее выгодно разработанный алгоритм размещения использовать при объеме данных 10000 Мб и количеством процессорных модулей от 75 до 100.

Было получена зависимость времени выполнения алгоритма размещения в тактах процессора. Результаты представлены на рисунке 8.

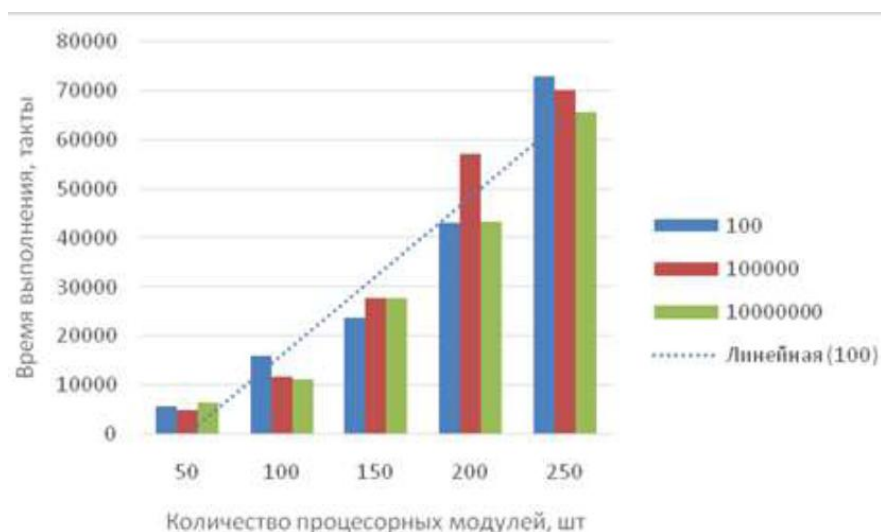


Рис. 8. Зависимость времени выполнения алгоритма от количества процессорных модулей

Каждый столбец гистограммы показывает объем данных, при котором проводилось моделирование. Синий – 100 Мб, оранжевый – 100000 Мб, серый – 10000000 Мб. По общей динамике видно, что с увеличением процессорных модулей, время выполнения алгоритма растет.

Если рассмотреть метод ветвей и границ, результаты которого представлены на рисунке 9, видно, что время выполнения этого метода измеряется порядком 10^9 степени тактов, в то время как разработанный в данной работе алгоритм – 10^4 тактов. Причем, значение времени растет линейно, а не экспоненциально.

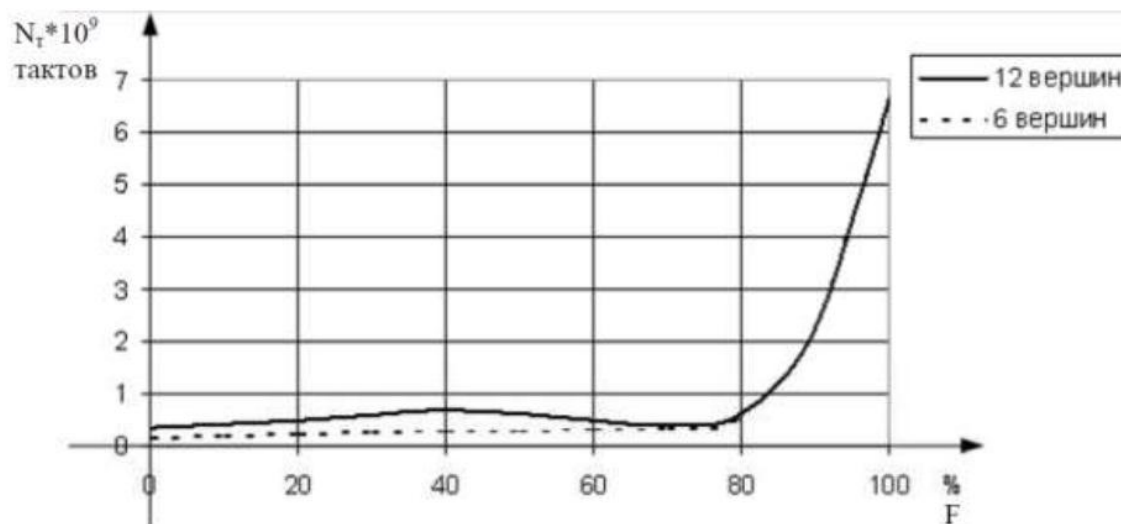


Рис. 9. Результаты размещения методом ветвей и границ

Выводы

В результате был предложен метод, методика и алгоритм беспроводной передачи данных в мультипроцессорных системах для перемещаемых объектов приема-передачи. Подход позволяет уменьшить суммарное значение коммуникационной задержки и увеличивать производительность мультипроцессорной системы. В дальнейших исследованиях будет проведена попытка аппаратной реализации предложенного в работе метода и алгоритма.

Предложенный метод и методика беспроводной передачи данных в мультипроцессорных системах для перемещаемых объектов обмена позволяет сделать вывод о возможности проектирования соответствующего алгоритма и специализированного устройства для использования в мультипроцессорных системах критического назначения. Это помогает уменьшить суммарное значение коммуникационной задержки и одновременного увеличения производительности мультипроцессорных систем.

Библиографический список

1. Гергель В.П. Теория и практика параллельных вычислений. - М.: Бином. Лаборатория знаний, 2007. – 424 с.
2. Цилькер Б.Я. Организация ЭВМ и систем. - СПб.: Питер, 2007. – 668 с.
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ Петербург, 2002. - 608 с.
4. Воеводин В.В. Вычислительная математика и структура алгоритмов. - М.: МГУ, 2006. - 112 с.
5. Хомоненко А.Д. Основы современных компьютерных технологий. – М.: Корона принт, 2009. – 448 с.
6. Закалюжный А.А. Развитие беспроводных сетей как средство контроля и управления удаленными системами // Молодой исследователь Дона. 2018. № 4 (13). С. 46 - 51.
7. Романов А.М., Гринголи Ф., Сикора А. Беспроводная синхронизация бортовых вычислительных устройств при помощи WiFi // Труды МАИ. 2019. № 108. URL: <http://trudymai.ru/published.php?ID=109522>
8. Оре О. Теория графов. - М.: Наука, 1968. - 352 с.
9. Колыбельников А.И. Обзор технологий беспроводных сетей // Труды МФТИ. 2012. Т. 4. № 2. С. 3 - 29.
10. Шевцов В.А., Бородин В.В., Крылов М.А. Построение совмещенной сети сотовой связи и самоорганизующейся сети с динамической структурой // Труды МАИ. 2016. № 85. URL: <http://trudymai.ru/published.php?ID=66417>

11. Бородин В.В., Петраков А.М., Шевцов В.А. Анализ эффективности передачи данных в сети связи группировки беспилотных летательных аппаратов // Труды МАИ. 2015. № 81. URL: <http://trudymai.ru/published.php?ID=57894>
12. Скачков М.К. Что такое Bluetooth и как он работает? URL: <http://1234g.ru/blog-of-wireless-technologies/about-bluetooth/что-такое-bluetooth-i-kak-on-rabotaet>
13. Технологии и стандарты Wi-Fi. Скорость интернета в Wi-Fi сети. URL: <http://wimax.livebusiness.ru/tags/WI-FI>
14. Борзов Д.Б., Масюков И.И. Планирование загрузки процессоров в мультипроцессорных системах критического назначения // Известия Юго-Западного государственного университета. 2018. № 6 (81). С. 168 - 175.
15. Борзов Д.Б. Аппаратные средства планирования размещения задач в мультипроцессорных системах критического назначения (теоретические основы): монография. – Курск: Юго-Западный государственный университет, 2018. - 179 с.
16. 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, 2015. 282 с.
17. Князева М.В. Метод ветвей и границ для решения задачи сетевого планирования с ограниченными ресурсами // Известия ЮФУ. Технические науки. 2010. № 7. С. 78 - 84.
18. Antamoshkin A.N., Kazakovtsev L.A. Random Search Algorithm for the p-Median Problem // Informatica, 2013, vol. 37 (3), pp. 267 - 278.

19. Морев Н.В. Сравнение алгоритмов планирования распределения задач для однородных распределенных вычислительных систем // Информационные технологии. 2010. № 5. С. 43 - 46.

20. Zhang L., Wong T.N. Solving integrated process planning and scheduling problem with constructive metaheuristics // Information Science, 2016, vol. 340 - 341, pp. 1 - 16.

DOI:[10.1016/j.ins.2016.01.001](https://doi.org/10.1016/j.ins.2016.01.001)

21. Zhang S., Wong, T.N. Integrated process planning and scheduling: An enhanced ant colony optimization heuristic with parameter tuning // Journal of Intelligent Manufacturing, 2018, vol. 29, pp. 585 – 601. URL: <https://doi.org/10.1007/s10845-014-1023-3>

22. Степина В.В. Архитектура ЭВМ и вычислительные системы. - М.: Курс Инфра-М, 2017. - 384 с.

23. Бородин В.В., Петраков А.М. Анализ алгоритмов управления адаптивной сетью передачи данных по локальным параметрам // Труды МАИ. 2015. № 80. URL: <http://trudymai.ru/published.php?ID=57035>

Method and technique of wireless data transmission in multiprocessor systems for non-stationary exchange objects

Borzov D.B.^{1*}, Dyubryuks S.A.^{1}, Sokolova Iu.V.^{2***}**

¹*Southwestern State University, SWSU, 94, str. 50 Let Oktyabrya, 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: serhio5551@yandex.ru

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

Abstract

The article considers data transmission improving in multiprocessor systems for movable objects using wireless data transmission between processor objects (software). The authors developed method and techniques for wireless data transmission with the ability to account for the movement in space of objects of exchange, which ensures increase in the multiprocessor systems performance and a reduction of the communication delay total value. With account for the task specifics, the authors perform the tasks place (programs, subroutines, files, etc.) between multiple software of a multiprocessor system, accounting for the current distance between the software, thus reducing the total communication delay, increasing the productivity and speed of the entire task performing en masse. The proposed ideas are supposed to be employed in critical multiprocessor systems (tracking, observation, aiming, control systems, etc.), when the extraordinary situation time in the system should be minimal. The proposed method and technique for wireless data transmission in multiprocessor systems for movable objects of exchange allows make conclusion on the possibility to design an appropriate algorithm and a

specialized device for application in critical purpose multiprocessor systems. It contributes to the reduction of the communication delay total value and concurrent increase in multiprocessor systems performance.

The authors accounted for the problem of computing systems (CS) compatibility and its architecture with the structure of the processed tasks, manifesting itself in the fact that any CS performs the assigned task only for a certain class of algorithms and methods. The assumption was made that the structure of the task and the architecture of the system did not correspond to each other, and the performance of the entire CS was minimal. As far as there is no structure that processes the tasks of various types equally well, several types of topologies, matching up a certain type of algorithm, are used while computing systems developing. Thus, both the CS topologies and methods of their accommodation should be considered within the framework of this work to achieve maximum performance. The authors of the presented work account for the fact that the CS is based on wireless technology. Thus, the performed analysis of such communication methods' protocols allowed revealing compatibility of their connection with topologies.

Keywords: wireless network, Wi-Fi, processor objects, information exchange algorithm, method and technique of wireless data transmission, non-stationary exchange objects, task placement.

References

1. Gergel' V.P. *Teoriya i ppaktika papallel'nykh vychiclenii* (Theory and practice of parallel calculations), Moscow, Binom. Labopatoriya znanii, 2007, 424 p.

2. Tsil'ker B.Ya. *Organizatsiya EVM i system* (Organization of computers and systems), Saint Petersburg, Piter, 2007, 668 p.
3. Voevodin V.V., Voevodin V.I. *Parallel'nye vychisleniya* (Parallel computations), Saint Petersburg, BKhV Peterburg, 2002, 608 p.
4. Voevodin V.V. *Vychislitel'naya matematika i struktura algoritmov* (Computational mathematics and structure of algorithms), Moscow, MGU, 2006, 112 p.
5. Khomonenko A.D. *Osnovy sovremennykh komp'yuternykh tekhnologii* (Modern computer technology basics), Moscow, Korona print, 2009, 448 p.
6. Zakalyuzhnyi A.A. *Molodoi issledovatel' Dona*, 2018, no. 4 (13), pp. 46 - 51.
7. Romanov A.M., Gringoli F., Sikora A. *Trudy MAI*, 2019, no. 108. URL: <http://trudymai.ru/eng/published.php?ID=109522>
8. Ore O. *Teoriya grafov* (Graph theory), Moscow, Nauka, 1968, 352 p.
9. Kolybel'nikov A.I. *Trudy MFTI*, 2012, vol. 4, no. 2, pp. 3 - 29.
10. Shevtsov V.A., Borodin V.V., Krylov M.A. *Trudy MAI*, 2016, no. 85. URL: <http://trudymai.ru/eng/published.php?ID=66417>
11. Borodin V.V., Petrakov A.M., Shevtsov V.A. *Trudy MAI*, 2015, no. 81. URL: <http://trudymai.ru/eng/published.php?ID=57894>
12. Skachkov M.K. *Chto takoe Bluetooth i kak on rabotaet?* URL: <http://1234g.ru/blog-of-wireless-technologies/about-bluetooth/chto-takoe-bluetooth-i-kak-on-rabotaet>
13. *Tekhnologii i standarty Wi-Fi. Skorost' interneta v Wi-Fi seti.* URL: <http://wimax.livebusiness.ru/tags/WI-FI>
14. Borzov D.B., Masyukov I.I. *Izvestiya Yugo-Zapadnogo gosudapstvennogo universiteta*, 2018, no. 6 (81), pp. 168 - 175.

15. Borzov D.B. *Apparatnye sredstva planirovaniya razmeshcheniya zadach v mul'tiprotsessornykh sistemakh kriticheskogo naznacheniya (teoreticheskie osnovy)* (Hardware tools for tasks scheduling in critical multiprocessor systems (theoretical basics)), Kursk, Yugo-Zapadnyi gosudarstvennyi universitet, 2018, 179 p.
16. Borzov D.B., Titov V.S. *Voprosy proektirovaniya i dinamicheskoi rekonfiguratsii topologii sistem logicheskogo upravleniya v sistemakh vysokoi gotovnosti: monografiya* (Issues of design and dynamic reconfiguration of of logical control systems topology in high availability systems: monograph), Kursk, Yugo-Zapadnyi gosudarstvennyi universitet, 2015, 282 p.
17. Knyazeva M.V. *Izvestiya YuFU. Tekhnicheskie nauki*, 2010, no. 7, pp. 78 - 84.
18. Antamoshkin A.N., Kazakovtsev L.A. Random Search Algorithm for the p-Median Problem, *Informatica*, 2013, vol. 37 (3), pp. 267 - 278.
19. Morev N.V. *Informatsionnye tekhnologii*, 2010, no. 5, pp. 43 - 46.
20. Zhang L., Wong T.N. Solving integrated process planning and scheduling problem with constructive metaheuristics, *Information Science*, 2016, vol. 340 - 341, pp. 1 - 16.
[DOI: 10.1016/j.ins.2016.01.001](https://doi.org/10.1016/j.ins.2016.01.001)
21. Zhang S., Wong, T.N. Integrated process planning and scheduling: An enhanced ant colony optimization heuristic with parameter tuning, *Journal of Intelligent Manufacturing*, 2018, vol. 29, pp. 585 – 601. URL: <https://doi.org/10.1007/s10845-014-1023-3>
22. Stepina V.V. *Arkhitektura EVM i vychislitel'nye sistemy* (Computer architecture and computing systems), Moscow, Kurs Infra-M, 2017, 384 p.
23. Borodin V.V., Petrakov A.M. *Trudy MAI*, 2015, no. 80. URL: <http://trudymai.ru/eng/published.php?ID=57035>