

СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

УДК 004.6

О.В. Фалеев

ПРИНЦИПЫ РЕАЛИЗАЦИИ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ИЗМЕРИТЕЛЬНОЙ ИНФОРМАЦИИ

Изложены принципы реализации параллельной обработки измерительной информации при построении сложных технических систем. Проведен анализ особенностей реализации параллельной обработки измерительной информации.

Ключевые слова: измерительная информация, параллельная обработка, конвейеризация.

Введение. Процесс обработки измерительной информации (ИИ) является многоуровневым, где с ростом уровня повышаются обобщающие свойства алгоритмов, приводящие, с одной стороны, к уменьшению потока данных, а с другой – к увеличению времени реализации уровневого алгоритма [1]. Большое значение имеет задача повышения производительности средств на самом низком уровне, алгоритмы обработки которого отличаются массовостью (т.е. обработка значительных объемов информации) и, как правило, требованием к оперативному получению результатов. Вместе с тем постановка задачи комплексной автоматизации испытаний сложных технических систем (СТС) требует повышения производительности на всех уровнях процесса обработки ИИ, начиная с подготовки к сеансу испытаний и кончая предоставлением результатов. Это приводит к выделению нескольких уровней организации параллельной обработки [2].

Тенденция распределенной обработки данных, полученных в результате проведения испытаний и экспериментов с СТС, требует построения систем автоматизации испытаний и экспериментов (САИЭ), как систем ПЭВМ, функционирующих параллельно. Вопрос о том, что должна представлять САИЭ в каждом конкретном случае (многомашинный комплекс, вычислительную сеть или распределенную систему), требует дополнительных исследований и должен решаться конкретно для условий применения [3].

Если САИЭ в целом должна представлять собой систему параллельно функционирующих ПЭВМ, то отдельные ее компоненты могут быть реализованы как параллельные ПЭВМ. Основным моментом при построении параллельных ПЭВМ является выбор принципа обработки, обеспечивающего высокую эффективность параллельных вычислений. Обусловлено это тем, что принцип параллельной обработки определяет структуру средств обработки, эффективность функционирования которых зависит от того, насколько эти средства приспособлены для реализации параллельной обработки ИИ. **Цель работы** – теоретический анализ особенностей реализации параллельной обработки ИИ.

Большинство алгоритмов улучшения качества ИИ, к которым относятся отбраковка единичных и групповых сбоев, автоконтроль, сглаживание, усреднение и сжатие [4], представляют собой рекурсии, т.е. последовательности вычислений, при которых значение последнего члена в последовательности зависит от одного или нескольких ранее вычисленных членов

$$y_i = f(x_i, y_{i-1}, y_{i-2}, \dots, y_0). \quad (1)$$

Вычисление рекурсии представляет собой определенную проблему, так как не обеспечиваются условия для параллельной обработки. Существуют методы введения параллелизма (векторизации), но они возможны только для случая линейной рекурсии первого порядка, когда

$$x_i = a_i \cdot x_{i-1} + b_i, \quad (2)$$

где a_i и b_i – константы. К этим методам относятся метод каскадных сумм и метод циклической редукции [5]. Реализация этих методов связана с маршрутизацией, т.е. с пересылками данных между процессорными элементами (ПЭ), что приводит к некоторому снижению производительности. В то же время наличие многоканальности средств измерения и независимость в большинстве случаев данных одних каналов от значений параметров других каналов являются предпосылкой организации параллельных вычислений с высокой эффективностью [6, 7]. В соответствии с классификацией параллелизма, приведенной в работе [5], параллельная обработка для многоканальной измерительной информации возможна либо параллельно по нескольким значениям одного параметра, либо по одному значению нескольких параметров параллельно.

Теоретическая часть. Вариант организации обработки путем параллельной обработки нескольких точек одного параметра будем рассматривать для группы алгоритмов типа «скользящего среднего» или усреднения

$$y_{cp} = \frac{1}{n} \cdot \sum_{i=1}^n y_i, \quad (3)$$

где функция y_i представляет собой общую линейную рекурсию первого порядка. Для простоты предположим, что обработка n точек одного параметра по алгоритму усреднения осуществляется на n ПЭ. На рисунке 1 показана маршрутизация алгоритма при использовании метода последовательных сумм.

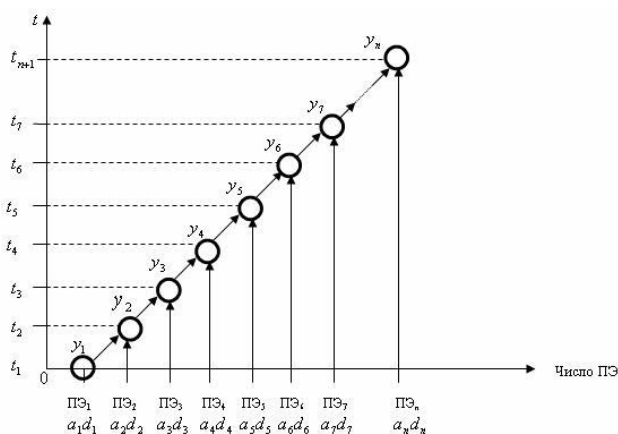


Рисунок 1 – Маршрутизация по методу последовательных сумм

Здесь кружочек обозначает вычисление процессорным элементом текущего значения функции y_{cp} , наклонная стрелка – пересылку результата из ПЭ_{*i*} в ПЭ_{*i+1*}, а вертикальная стрелка указывает на константы a_i и d_i , участвующие в вы-

числениях $y_{cp i}$ в ПЭ_{*i*}. Из рисунка видно, что в момент времени t_i вычисления осуществляются только в ПЭ_{*i*}, а пересылка осуществляется только из одного ПЭ, в данном случае из ПЭ_{*i*}. Такой метод требует n вычислений со степенью параллелизма, равной 1, и $(n-1)$ пересылок со степенью параллелизма, равной 1. Время реализации алгоритма в этом случае составит:

$$T_p = n \cdot \tau_{обр} + (n-1) \cdot \tau_{пер}, \quad (4)$$

где $\tau_{обр}$ – время вычисления y_i в ПЭ_{*i*}, а $\tau_{пер}$ – время пересылки между ПЭ_{*i*} и ПЭ_{*i-1*}. Эффективность параллелизма в таком случае определяется как

$$\varepsilon = \frac{T_u}{T_p} = \frac{\tau_{обр}}{n \cdot \tau_{обр} + (n-1) \cdot \tau_{пер}} = \frac{1}{n + (n-1) \cdot \gamma}, \quad (5)$$

где T_u – идеальное время реализации алгоритма без пересылок при максимальной степени параллелизма обработки;

$$\gamma = \frac{\tau_{пер}}{\tau_{обр}}.$$

В случае реализации алгоритма усреднения по методу каскадных сумм организация вычислений имеет вид, показанный на рисунке 2.

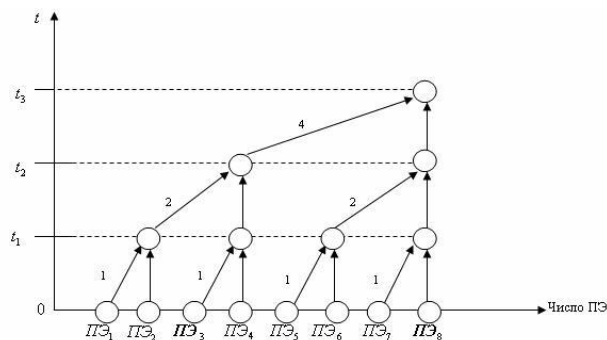


Рисунок 2 – Маршрутизация по методу каскадных сумм

Метод каскадных сумм используется при $x_0 = 0$. Этим методом достаточно просто реализуется сумма в выражении (3), если y_i представляют собой константы, а не функции. Для простоты изложения ограничимся $n = 8$. На рисунке 3 видно, что число шагов равно $\log_2 n$, на каждом i – м шаге реализуются вычисления со степенью параллелизма $n \cdot 2^{-i}$ и осуществляются пересылки длиной i (предполагаем, что пересылка из ПЭ_{*i-2*} в ПЭ_{*i*} эквивалентна двум пересылкам из ПЭ_{*i-1*} в ПЭ_{*i*} и т.д.) со степенью параллелизма $n \cdot 2^{-i}$.

Время реализации алгоритма усреднения методом каскадных сумм равно

$$T_p = \tau_{обр} \cdot \log_2 n + \tau_{пер} \cdot \sum_{j=1}^{\log_2 n} 2^{j-1}. \quad (6)$$

После преобразования выражения (6) имеем

$$T_p = \tau_{обр} \cdot \log_2 n + (n-1) \cdot \tau_{пер}. \quad (7)$$

Эффективность параллелизма при использовании метода каскадных сумм равна

$$\begin{aligned} \varepsilon &= \frac{T_u}{T_p} = \frac{\tau_{обр}}{\tau_{обр} \cdot \log_2 n + (n-1) \cdot \tau_{пер}} = \\ &= \frac{1}{\log_2 n + (n-1)\gamma}. \end{aligned} \quad (8)$$

Эффективность параллелизма при использовании метода каскадных сумм может быть увеличена, в частности, за счет более рациональной организации маршрутизации. Анализ маршрутизации, использующей регулярные каналы пересылок между ПЭ, показывает, что на более высоких шагах схемы реализации алгоритма имеются свободные ПЭ, которые находятся между двумя ПЭ, пересылающими данные. Использование этих свободных ПЭ позволит сократить длину пересылок, но при этом пересылки должны осуществляться в обоих направлениях, т.е. эффективность параллелизма прямо зависит от степени использования (загрузки) ПЭ. На рисунке 3 показан способ маршрутизации вычислений с повышенной загрузкой ПЭ.

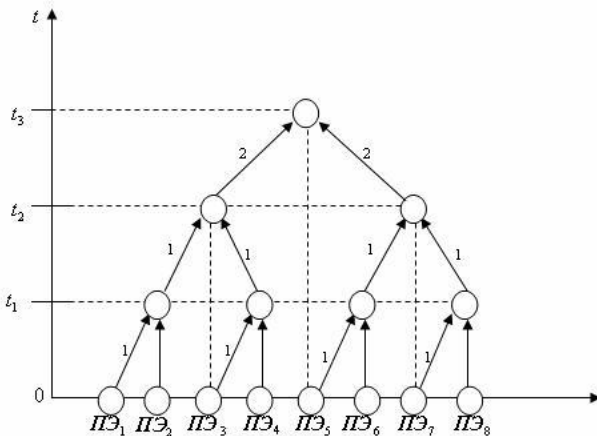


Рисунок 3 – Маршрутизация вычисления рекурсии при рациональной организации пересылок

Время реализации алгоритма вычисления рекурсии при повышенной загрузке ПЭ равно

$$T_p = \tau_{обр} \cdot \log_2 n + \tau_{пер} \cdot \left(\sum_{j=1}^{\log_2 n} 2^{j-1} - \sum_{j=2}^{\log_2 n} 2^{j-2} \right) \cdot \tau_{пер}. \quad (9)$$

После преобразования выражения (9) имеем:

$$T_p = \tau_{обр} \cdot \log_2 n + \left(1 + \sum_{j=0}^{\log_2(n-2)} 2^j\right) \tau_{пер} = \tau_{обр} \cdot \log_2 n + \frac{n}{2} \cdot \tau_{пер}. \quad (10)$$

Эффективность параллелизма в этом случае равна

$$\varepsilon = \frac{\tau_{обр}}{\tau_{обр} \cdot \log_2 n + \frac{n}{2} \cdot \tau_{пер}} = \frac{1}{\log_2 n + \frac{n}{2} \cdot \gamma}. \quad (11)$$

Обобщением метода каскадных сумм для случая $x_0 \neq 0$ является метод циклической редукции, основная идея которого заключается в объединении смежных членов рекурсии таким образом, чтобы выявить зависимость между каждым предыдущим и последующим членами рекурсии. Эта зависимость также представляет собой линейную рекурсию первого порядка, хотя коэффициенты уже имеют другой смысл, чем в выражении (2). Этот процесс можно повторить для получения рекуррентных формул, связывающих каждый четвертый член, каждый восьмой член и т.д. Свое название метод получил из-за того, что при его реализации на последовательной ПЭВМ число используемых уравнений рекурсии на каждом последующем шаге уменьшается вдвое. Реализация метода циклической редукции на параллельной ПЭВМ требует поддержания параллелизма на высоком уровне, поэтому число уравнений искусственно не уменьшается. Для двух последующих членов x_{i-1} и x_i можно записать:

$$x_i = a_i \cdot x_{i-1} + b_i, \quad (12)$$

$$x_{i-1} = a_{i-1} \cdot x_{i-2} + b_{i-1}, \quad (13)$$

$$x_i = a_i \cdot a_{i-1} \cdot x_{i-2} + a_i \cdot b_{i-1} + b_i. \quad (14)$$

Обозначив

$$a_i \cdot a_{i-1} = a_i^{(1)}, \quad (15)$$

$$a_i \cdot b_{i-1} + b_i = b_i^{(1)}, \quad (16)$$

получим:

$$x_i = a_i^{(1)} \cdot x_{i-2} + b_i^{(1)}. \quad (17)$$

Применение этого метода для выявления зависимости каждого четвертого, каждого восьмого и т.д. членов для уровня $m (0, 1, \dots, \log_2 n)$ дает:

$$x_i = a_i^{(m)} \cdot x_{i-2^m} + b_i^{(m)}, \quad (18)$$

где

$$a_i^{(m)} = a_i^{(m-1)} \cdot a_{i-2^{m-1}}^{(m-1)} \quad (19)$$

$$b_i^{(m)} = a_i^{(m-1)} \cdot b_{i-2^{m-1}}^{(m-1)} + b_i^{(m-1)} \quad (20)$$

и начальные значения

$$a_i^{(0)} = a_i, b_i^{(0)} = b_i. \quad (21)$$

Индекс m в скобках указывает на номер уровня выполнения алгоритма. Схема реализации метода циклической редукции показана на рисунке 4.

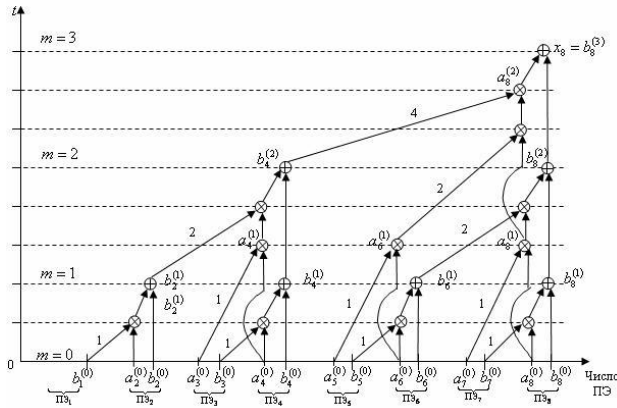


Рисунок 4 – Организация вычисления рекурсий методом циклической редукции

Знак в кружочке на схеме указывает выполняемую арифметическую операцию. Время реализации алгоритма на уровне m равно

$$T_p = l(m-1) \cdot \tau_{умн} + \tau_{умн} + \tau_{сл} + 2^{m-2} \cdot \tau_{пер} + 2^{m-1} \cdot \tau_{пер}, \quad (22)$$

где $l(m-1)$ – единичная функция, равная

$$l(m-1) = \begin{cases} 1, & \text{если } m > 1 \\ 0, & \text{если } m \leq 1 \end{cases} \quad (23)$$

$\tau_{умн}$ – время выполнения операции умножения;

$\tau_{сл}$ – время выполнения операции сложения.

Суммируя по всем уровням, получаем:

$$T_p = \sum_{m=1}^{\log_2 n} T_p^{(m)} = \sum_{m=1}^{\log_2 n} \{ \tau_{умн} \cdot [1 + l(m-1)] + \tau_{сл} + 3 \cdot 2^{m-2} \cdot \tau_{пер} \}. \quad (24)$$

После преобразования получаем:

$$T_p = (2 \log_2 n - 1) \cdot \tau_{умн} + \tau_{сл} \cdot \log_2 n + 3 \frac{n}{2} \cdot \tau_{пер}. \quad (25)$$

Эффективность параллелизма в этом случае равна

$$\varepsilon = \frac{\tau_{умн} + \tau_{сл}}{\log_2 n \left[\left(2 - \frac{1}{\log_2 n} \right) \cdot \tau_{умн} + \tau_{сл} + \frac{3n \cdot \tau_{пер}}{2 \log_2 n} \right]}. \quad (26)$$

Обозначив через $\tau_{обр}$ сумму $(\tau_{умн} + \tau_{сл})$ (т.к. каждый ПЭ выполняет эти операции для получения промежуточных результатов), имеем:

$$\varepsilon = \frac{1}{\log_2 n \left(1 + \frac{3n}{2 \log_2 n} \cdot \gamma_1 - \frac{\alpha}{\log_2 n} \right)} = \frac{1}{\log_2 n + \frac{3n\gamma_1}{2} - \alpha}, \quad (27)$$

где

$$\begin{cases} \gamma_1 = \frac{\tau_{пер}}{2\tau_{умн} + \tau_{сл}} \\ \alpha = \frac{\tau_{умн}}{2\tau_{умн} + \tau_{сл}} \end{cases}. \quad (28)$$

Использование метода пересылок в обоих направлениях, позволяет сократить время реализации алгоритма. На рисунке 5 приведена схема такой организации вычислений.

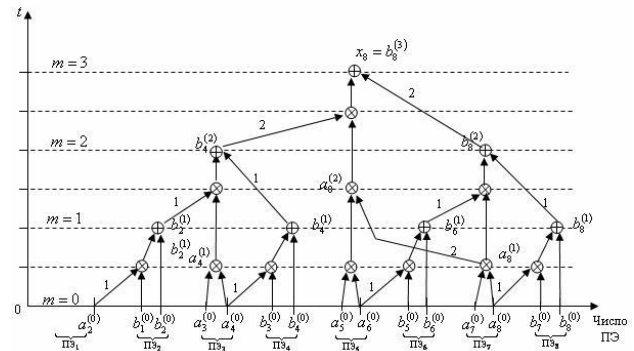


Рисунок 5 – Улучшенная организация вычисления рекурсий методом циклической редукции

В этом случае время выполнения алгоритма определяется как

$$T_p = (\tau_{сл} + \tau_{умн}) \cdot \log_2 n + (n-1) \cdot \tau_{пер}, \quad (29)$$

а эффективность параллелизма

$$\varepsilon = \frac{(2\tau_{умн} + \tau_{сл})}{(\tau_{умн} + \tau_{сл}) \cdot \log_2 n + (n-1) \cdot \tau_{пер}} = \frac{1}{(\alpha + \beta) \cdot \log_2 n + (n-1) \cdot \gamma_1}, \quad (30)$$

где

$$\beta = \frac{\tau_{сл}}{2\tau_{умн} + \tau_{сл}}. \quad (31)$$

Анализ вариантов организации обработки нескольких точек одного параметра указывает на малую загрузку ПЭ и, как следствие этого, достаточно низкую эффективность параллелизма. Реально эффективность может оказаться еще ниже из-за последовательного характера потока измерений одного параметра. Для повышения

эффективности необходимо уменьшать число пересылок и увеличивать степень параллелизма. А при реализации алгоритма циклической редукции некоторые ПЭ простаивают только из-за того, что приходится ожидать нужной операции.

При испытаниях СТС возникает задача обработки данных нескольких однотипных датчиков, например давления, установленных в различных точках этой системы, по одному и тому же алгоритму.

Наиболее просто такая обработка реализуется при использовании метода последовательных сумм. Организация вычислений при этом имеет вид, показанный на рисунке 6.

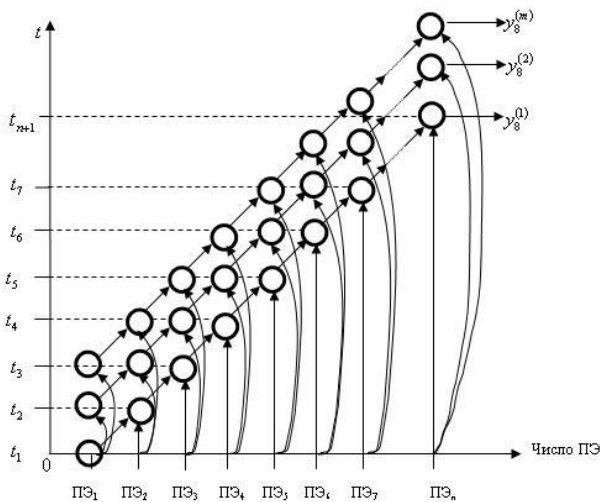


Рисунок 6 – Параллельная обработка нескольких параметров методом последовательных сумм

На каждом уровне процесса выполняется обработка со степенью параллелизма, равной номеру уровня по mod n , и единичная пересылка той же степени параллелизма. После уровня n и до уровня m степень параллелизма обработки и пересылок равна n . Для уровней, больших m , где m – число обрабатываемых параметров, степень параллелизма уменьшается на единицу. Суммарное время обработки m параметров равно

$$T_p(m) = (m + n - 1) \cdot \tau_{обр} + (m + n - 2) \cdot \tau_{пер}, \quad (32)$$

а среднее время вычисления

$$T_p = \frac{m + n - 1}{m} \cdot \tau_{обр} + \frac{m + n - 2}{m} \cdot \tau_{пер}. \quad (33)$$

Эффективность параллелизма равна

$$\varepsilon = \frac{m \cdot \tau_{обр}}{[(m + n - 1) \cdot \tau_{обр} + (m + n - 2) \cdot \tau_{пер}]}. \quad (34)$$

При параллельной обработке нескольких параметров последовательно точка за точкой можно за каждым ПЭ закрепить один параметр.

Среднее время обработки n параметров за m шагов равно

$$T_p = \frac{m \cdot \tau_{обр}}{n}. \quad (35)$$

Выводы. Выше рассматривалась параллельная обработка одного n -мерного вектора. Между тем параллельная обработка нескольких векторов позволит еще более повысить эффективность параллелизма, что наглядно иллюстрирует сравнение выражений (15) и (34). Конвейерно-параллельная организация обработки наиболее полно соответствует последовательному характеру потока данных. Конвейеризация позволяет более полно использовать производительность каждого ПЭ. Из рисунков 1 – 5 видно, что некоторые ПЭ не полностью загружены. Однако наибольший эффект конвейеризация дает при одинаковой загрузке ПЭ. По этой причине использование конвейеризации при вычислении рекурсии по немодифицированному методу каскадных сумм и методу циклической редукции не дает никакого эффекта, так как восьмой ПЭ занят все время, а остальные ПЭ загружены неравномерно. С точки зрения эффективности конвейеризации при вычислении рекурсии наиболее перспективен модифицированный метод каскадных сумм, так как обеспечивает равномерную загрузку каждого ПЭ (не более чем на шаг за цикл обработки). Это позволяет загрузить ПЭ обработкой следующего параметра, не дожидаясь полного окончания обработки текущего параметра. Пространственно-временной граф организации конвейерной обработки с использованием модифицированного метода каскадных сумм показан на рисунке 7.

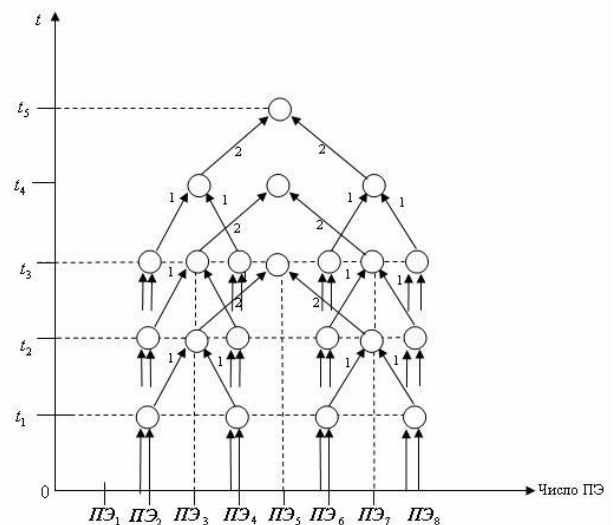


Рисунок 7 – Конвейеризация вычисления рекурсии при использовании модифицированного метода каскадных сумм

Для уменьшения количества пересылок на первом шаге используется загрузка каждого четного ПЭ не только своими данными, но и данными предыдущего ПЭ. Начиная с шага $t_{\log_2 n}$ и до шага t_m степень параллелизма обработки равна $(n-1)$. А пересылки на шаге t_2 – единичные, а с шага t_2 и до шага t_{m+2} – двойные.

При обработке n -мерных векторов начиная с шага $t_{\log_2 n-1}$ и до шага $t_{m-1+\log_2 n}$ величина пересылок равна $\frac{n}{4}$. В интервале шагов от t_1 до $t_{\log_2 n-2}$ размер пересылок равен 2^{j-1} , $j=1, \log_2 n-2$. Суммарное время обработки равно

$$T_p(m) = \tau_{обр} \cdot (m-1 + \log_2 n) + \frac{mn}{4} \cdot \tau_{пер} + \tau_{пер} \cdot \sum_{j=1}^{\log_2 n-2} 2^{j-1} = (m-1 + \log_2 n) \cdot \tau_{обр} + \frac{n(m+1)-4}{4m} \cdot \tau_{пер}. \quad (36)$$

Среднее время обработки одного параметра составляет:

$$T_p = \left(1 + \frac{\log_2 n}{m}\right) \cdot \tau_{обр} + \frac{n(m+1)-4}{4m} \cdot \tau_{пер}. \quad (37)$$

Эффективность параллелизма такой конвейеризации равна

$$\varepsilon = \frac{m \tau_{обр}}{(m-1 + \log_2 n) \tau_{обр} + \frac{n(m+1)-4}{4} \tau_{пер}} = \frac{4m}{4(m-1 + \log_2 n) + (n+mn-4)\gamma}. \quad (38)$$

Закключение. В статье изложены принципы реализации параллельной обработки измерительной информации при построении сложных технических систем. Проведен анализ особенностей реализации параллельной обработки измерительной информации. Определено преимущество конвейерно-параллельной организации обработки.

Библиографический список

1. Лебедев Г.Н. Методы обработки измерительных потоков. – М.: МАИ, 1983. – 48 с.
2. Лацис А.О. Параллельная обработка данных. – М.: Академия, 2010. – 336 с.
3. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. – СПб.: Питер, 2003. – 877 с.
4. Романов В.Н. Теория измерений. Методы обработки результатов измерений. – СПб.: СЗТУ, 2006. – 127 с.
5. Хокни Р., Джесссхоуп К. Параллельные ПЭВМ: пер. с англ. – М.: Радио и связь, 1986. – 392 с.
6. Гергель В.П. Теория и практика параллельных вычислений. – М.: Бином, 2007. – 424 с.
7. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.