



Рис. 10. Радиопульсы на входе (верхняя эюра) и выходе (нижняя эюра) цепи емкостной связи

Таким образом, используя крупногабаритные конденсаторы с вращающимися обкладками можно осуществить многоканальную передачу информации между подвижными объектами.

С помощью предложенной модели блока крупногабаритных конденсаторов доказано, что варьируя конструкцию блока конденсаторов, можно осуществить передачу широкой номенклатуры сигналов, исключая постоянный ток, используемых в технических устройствах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гордиенко В.Н., Творецкий М.С. Многоканальные телекоммуникационные системы. -М.: Горячая линия – Телеком, 2005. -416 с.
2. Многоканальные системы передачи /Под. ред. Н.Н. Басовой и В.Н. Гордиенко – М.: Радио и связь, 1997.- 560 с.

УДК 681.518

В.А. Литвиненко, С.А. Ховансков

РЕШЕНИЯ ЗАДАЧ ПУТЕМ ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В СЕТИ

При решении задач, требующих обработки большого объема данных, возникает проблема получения решения за приемлемое время. Одним из путей выполнения ограничения на временной ресурс - разделение всего объема вычислений между несколькими центрами обработки данных [1]. Обычно в качестве центров обработки стараются использовать работающие совместно персональные компьютеры.

При организации совместной работы компьютеров неизбежна проблема разделения задачи на вычислительные блоки между центрами обработки. Традиционно она решается созданием управляющей программы. Управляющая программа закрепляет между компьютерами вычислительные блоки задачи, организывает их совместную работу, следит за работоспособностью компьютеров, обеспечивает сбор обработанных данных [2].

Такой подход накладывает жесткие ограничения на круг выполняемых задач из-за невозможности быстрой перенастройки компьютеров в случае необходимо-

сти выполнения ряда задач с разными вычислительными алгоритмами, Также недостатками различных реализаций такого подхода являются неизбежные временные потери, большие затраты на разработку программы и низкая живучесть системы из-за наличия центра управления, низкая эффективность решения вычислительных блоков задач с высокой степенью связности по данным.

В настоящее время активно развивается направление, связанное с организацией процесса принятия решения путем распределения объемов решаемой задачи с помощью технологии мобильных агентов, называемой мультиагентной системой. Достоинствами технологии мобильных агентов является высокая надежность и возможность поддержки независимых вычислительных процессов [3].

В статье предлагается алгоритм работы мультиагентной системы, позволяющий организовать распределенные вычисления на нескольких персональных компьютерах (ПК) методом принятия агентами коллективного решения.

При организации распределенных вычислений система решает проблему конфигурирования вычислительного процесса в компьютерной среде. Предложенный алгоритм предусматривает не только организацию совместной работы имеющихся центров обработки информации, но и предпринимает попытки увеличения количества центров обработки данных, оптимизирует распределение вычислительных блоков в ходе выполнения компьютерами задачи с целью сокращения общего времени решения и повышения степени живучести всей системы

Разработанный алгоритм работы мультиагентной системы позволяет использовать в качестве центров обработки не специализированные машины с необходимым аппаратным обеспечением, а обычные ПК, что уменьшает затраты и увеличивает живучесть вычислительного процесса.. В организуемой системе нет требований к быстродействию компьютеров, а также не существует ограничение на количество ПК. Необходимым условием является наличие физических связей между ПК для передачи данных.

Это может быть и обычная компьютерная сеть, построенная по топологии, имеющей иерархическую структуру, использующей различные сетевые технологии (например Ethernet). При такой конфигурации и сетевой технологии каждый ПК может обращаться к источнику входных данных по каналу связи независимо от других ПК, передавать данные и служебную информацию от компьютера любому другому компьютеру.

Сеть используется для получения обрабатываемой информации (входные данные) из одного источника во все ПК (например сеть Интернет, решение задач САПР, задачи управления и др.). Все результаты решений передаются по сети и собираются на одном из ПК.

Модуль мультиагентной системы (агент) - это одна и та же программа, устанавливаемая на каждый компьютер. Включает в себя вычислительные модули, необходимые для выполнения самой задачи обработки данных и управляющую программу – агент, реализующую предлагаемый алгоритм. Агент обладает правом использовать все вычислительные ресурсы своего ПК, находящегося в сети. При использовании ПК аппаратные ресурсы и маршруты передачи данных между агентами образуют иерархическую вычислительную структуру. Входные данные поступают агентам всех уровней структуры, а выходные данные собираются у агента самого верхнего уровня. Агент самого верхнего уровня располагается на ПК пользователя, получающего результаты решения задачи.

Первоначально модуль агента может располагаться на одном или нескольких ПК. Все агенты являются копиями одного агента и работают по одному и тому же алгоритму. Агенты равноправны и управляют работой своего компьютера независимо друг от друга. Под управлением своего агента ПК занимается сбором и обра-

боткой информации. По окончании выполнения заданного объема вычислений выполняется обмен между агентами ПК, имеющими связи согласно иерархической структуре.

В ходе работы каждый агент старается увеличить количество агентов в сети и начинает поиск свободных от других агентов ПК. При обнаружении таких ПК выполняется их «вербовка». При этом происходит передача от агента-родителя своей копии в виде управляющего модуля агента-клона, который будет работать на «завербованном» ПК. В случае передачи от агента-клона обработанных данных адрес ПК запоминается как агент более низкого уровня. В случае отсутствия передачи ПК адрес запоминается как занятый. Появившийся агент-клон в иерархической структуре будет принадлежать нижнему уровню мультиагентной системы по отношению к уровню агента-родителя.

Каждый агент-клон обменивается выходными данными только с тем агентом, который для него является родителем. Разрастаясь таким путем, иерархическая структура вычислительной системы может иметь неограниченное количество уровней.

Агент-клон является полной копией породившего его агента-родителя и будет выполнять те же функции, что и агент-родитель – управление обработкой данных, обмен ею с другими агентами, передача служебной информации агентам-родителям и своим агентам-клонам, поиск и вербовка новых свободных ПК. Количество агентов-клонов у порождаемого их агента-родителя может быть бесконечно.

Все множество агентов делится на совокупности агентов-родителей и их агентов-клонов

$$I_{\beta}^{i+1}, I_{\beta+1}^{i+1}, \dots, I_{\beta+(k-1)}^{i+1} \in I_{\alpha}^i, \quad (1)$$

где I_{α}^i - ПК на i -ом уровне, на котором расположен агент-родитель.

Каждый агент-родитель, имея агентов-клонов, передает им вместе с служебной информацией сведения, содержащие данные о наиболее коротком маршруте передачи выходных данных и служебной информации от агента-клона агенту самого высокого уровня.

Для повышения живучести системы информация о маршруте используется в случае выхода из строя или отключения ПК, входящих в маршрут передачи данных. Если из указанного маршрута выбыл или отключился ПК, то связанный с ним агент-клон более нижнего уровня передает информацию агенту ближайшего подключенного ПК высокого уровня, присутствующего в указанном маршруте передачи результата.

При многоуровневой иерархии агент-клон может быть одновременно и агентом-родителем.

При работе агентов весь процесс работы мультиагентной системы делится на одинаковые по действиям каждого агента шаги. Под шагом понимается повторяющийся набор действий: прием данных, их обработка, прием и обработка служебной информации, ее передача и передача своего и принятого результатов.

Шаг начинается с того, что после обработки данных каждый агент-клон передает агенту, являющимся его родителем, помимо выходных данных, еще и служебную информацию о ПК, на которых эти вычисления выполнялись:

- скорость выполнения данным ПК заданного объема вычислений;
- время получения выходных данных и передачи;
- время приема обработанных данных.

Это «прямая» связь в мультиагентной системе снизу-вверх, позволяет осуществить сбор результатов вычислений при любом количестве центров обработки данных.

Существует и «обратная» связь сверху-вниз. Она выполняется после передачи информации по «прямой» связи. Все агенты-клоны после своей передачи информации по прямой связи ждут передачи служебной информации от своих агентов-родителей.

После получения результатов от агентов нижнего уровня агентами-родителями анализируются служебные данные, полученные при «прямой» передаче. При «обратной» передаче от агента-родителя высокого уровня всем связанным с ним агентам-клонам нижнего уровня будет передана служебная информация, содержащая, если это нужно, управляющие данные о назначении агенту-клону объема вычислений выполняемых при обработке входных данных, и путь передачи выходных данных на самый верхний уровень.

Каждый агент обменивается на каждом шаге информацией с агентом верхнего уровня иерархической структуры, являющимся его родителем, и своими агентами-клонами нижнего уровня, которые поддерживают с ним связь. По этим связям агент занимается не только приемом и передачей обработанных данных на верхний уровень, но и формированием и передачей служебной информации на верхний и нижний уровни структуры. Для агента время шага складывается из времени приема входных данных, выполнения своего объема вычислений, времени обработки и передачи различной информации агентам соседних уровней.

Основной задачей для агента остается выполнение своего объема вычислений решаемой задачи, какой бы уровень он ни занимал.

Агент-родитель при появлении своего нового агента-клона пытается перераспределить весь объем обрабатываемых данных, который назначен ему агентом-родителем на части примерно в равной пропорции между своими агентами-клонами, с учетом своих собственных вычислительных ресурсов. При последующих шагах происходит корректировка объема обрабатываемых данных им и его клонами с учетом вычислительных ресурсов каждой ПК. Сам агент-клон, являясь агентом-родителем, также изменяет объем заданий своим агентам-клонам в соответствии с установленной от своего агента-родителя долей от общего объема обрабатываемых данных.

Это позволяет уменьшить разницу во времени получения результатов решения на всех ПК мультиагентной системы одного уровня и повышает надежность работы системы. Выравнивание может выполняться и по другим критериям в зависимости от объема вычислений и пропускной способности каналов связи.

За P можно принять весь объем обрабатываемой информации на одном шаге. В самом начале работы мультиагентной системы структура имеет один уровень и агент работает только на одном ПК $n1 = 1$. Соответственно весь объем вычислений P выполняется агентом на этом ПК. При расширении структуры каждый ПК присоединяется к соответствующему уровню $l_\alpha \in n1$. Агенты второго уровня образуют множество $n2 = (l_1^2, l_2^2, \dots, l_m^2)$.

При размножении агента на другие ПК весь объем вычислений (2) начинает делиться между всеми агентами мультиагентной системы.

$$P_\alpha^i = P_\beta^{i+1} + P_{\beta+1}^{i+1} + \dots + P_{\beta+(k-1)}^{i+1} + P_\alpha^i, \quad (2)$$

где k – количество агентов-клонов агента-родителя ПК l_α^i ,

Время решения задачи на одном шаге (3) определяется наибольшим временем выполнения своего объема вычислений этого шага одним из агентов, входящих в совокупность, агент-родитель и его агенты-клоны.

$$t_{\alpha}^i = \max(t_{\beta}^{i+1}, t_{\beta+1}^{i+1}, \dots, t_{\beta+(k-1)}^{i+1}). \quad (3)$$

Для второго уровня это будет

$$t^1 = \max(t_1^2, t_2^2, \dots, t_m^2). \quad (4)$$

Среднее время вычислений (5) агентами-клонами будет

$$t_{cp\alpha}^i = t_{\beta}^{i+1} + t_{\beta+1}^{i+1} + \dots + t_{\beta+(k-1)}^{i+1} = \frac{\sum_{j=0}^{j=k-1} t_{m+j}^{i+1}}{k}. \quad (5)$$

Агент-родитель перераспределяет между своими агентами клонами объемы производимых вычислений (6) для уменьшения максимального времени выполнения агентами-клонами своих задач

$$(P_m^{i+1})' = P_m^{i+1} \frac{t_{cp\alpha}^i}{t_m^{i+1}}. \quad (6)$$

Алгоритм агента

1. Получение входных данных.
2. Запуск выполнения обработки входных данных.
3. Адрес присутствующего в сети компьютера есть в списке проверенных компьютеров? Да, перейти к п.5, нет - п.4.
4. Передача обнаруженному компьютеру копии агента.
5. Выполнена обработка входных данных? Да, перейти к п.6, нет - п.3.
6. Получены выходные данные? Да, перейти к п.7, нет - п.3.
7. Запись полученных данных.
8. Получены данные от агентов уровня $i - 1$? Да, перейти к п.9, нет - п.10.
9. Занесение компьютера-отправителя в список агентов уровня $i - 1$.
10. Все агенты уровня $i - 1$ передали данные? Да, перейти к п.11, нет - п.3.
11. Передача всех выходных данных.
12. Получена служебная информация от агента уровня $i + 1$? Да, перейти к п.14, нет - п.13.
13. Передача всех выходных данных следующему агенту согласно маршрута передачи данных.
14. Расчет $t_{cp\alpha}^i$ и P_m^{i+1} '.
15. Передача служебной информации агентам уровня $i - 1$.
16. Переход к п.1.

Разработанный алгоритм организует и оптимизирует распределенные вычисления методом принятия коллективного решения, с использованием мультиагентной системы и выполняется на фоне решения основной задачи. Время, затраченное

на переконфигурацию структуры мультиагентной системы, из-за изменений в системе или поступления нового задания сводится к минимуму и оказывает относительно небольшое влияние на общее время решения задачи.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лорин Г. Распределенные вычислительные системы.// -М.: Радио и связь, 1984. - 296 с.
2. Таненбаум Э.. Распределенные системы: принципы и парадигмы.// -СПб: Питер, 2003. – 877 с.
3. Ховансков С.А., Мельник Э.В., Блушвили И.В. Метод организации распределенных вычислений в управляющих системах // Мехатроника, автоматизация, управление. «Новые технологии», N4. 2003.

УДК 621.391.25(075)

И.Л. Трунов, Т.М. Горягина

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ НА ОСНОВЕ КОДОВ С ИРРАЦИОНАЛЬНЫМ ОСНОВАНИЕМ

Помехоустойчивое кодирование передаваемой информации позволяет в приемной части системы обнаруживать и исправлять ошибки. Коды, применяемые при помехоустойчивом кодировании, называются корректирующими кодами.

Как правило, корректирующий код может исправлять меньше ошибок, чем обнаруживать. Число ошибок, которые корректирующий код может исправить в определенном интервале последовательности двоичных символов, например, в одной кодовой комбинации, называется исправляющей способностью кода.

Существует система счисления с иррациональным основанием $\tau = \frac{1 + \sqrt{5}}{2}$,

получившая название системы счисления Фибоначчи в честь итальянского математика 13-го столетия Леонардо Фибоначчи [1]. Эта система обладает рядом свойств, на основе которых можно построить помехоустойчивые коды.

Данная работа посвящена анализу помехоустойчивости кодов с иррациональным основанием (кодов Фибоначчи), а также построению комплекса помехоустойчивого кодирования для систем связи.

Под r -кодом Фибоначчи понимается следующий способ представления натурального числа N :

$$N = a(n)F_p(n) + a(n-1)F_p(n-1) + \dots + a(i)F_p(i) + \dots + a_1F_p(1), \quad (1)$$

где $a(n) = \{0, 1\}$ - двоичная цифра i -го разряда представления; n - разрядность представления; $F_p(i)$ - r -число Фибоначчи, задаваемое с помощью следующих рекуррентных формул:

$$F_p(i) = F_p(i-1) + F_p(i-p-1); \quad (2)$$

$$F_p(1) = F_p(2) = \dots = F_p(p+1) = 1, \quad (3)$$