

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Воробьев Н.Н.* Числа Фибоначчи. – М.: Наука, 1978. – 144 с.
2. *Стахов А.П.* Кодирование данных, основанное на фибоначчиевых матрицах // Тр. Международной конференции «Проблемы гармонии, симметрии и золотого сечения в природе, науке и искусстве». – Винница: Изд-во Винницкого государственного аграрного университета. – 2003. – С. 311-325.
3. *Румянцев К.Е., Трунов И.Л., Горягина Т.М.* Система помехоустойчивого кодирования с переменной избыточностью // Электротехнические и информационные комплексы и системы. – 2007. – Т. 3, № 3. – С. 55-58.
4. *Ковалев А.Р., Кулагина М.И., Трунов И.Л.* Система помехоустойчивого кодирования на основе второй р-последовательности Фибоначчи // Информационное противодействие угрозам терроризма. – 2010. – № 13. – С. 139-143.

Статью рекомендовал к опубликованию к.т.н. Е.А. Семерников.

Трунов Игорь Леонидович – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: igortil@mail.ru; 347923, г. Таганрог, ул. Прохладная, 5, кв. 22; тел.: +79185384139; к.т.н.; доцент.

Линенко Ульяна Демьяновна – e-mail: uliana555555@mail.ru; г. Таганрог, ул. Чехова, 333, кв. 170; тел.: +79043459313; студентка.

Пустоварова Александра Витальевна – e-mail: 000shurik000@rambler.ru; г. Таганрог, ул. Чехова, 22, кв. 409; тел.: 89885898232; студентка.

Trunov Igor Leonidovich – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: igortil@mail.ru; 5, Prohladnaya street, ap. 22, Taganrog, 347923, Russia; phone: +79185384139; cand. of eng. sc.; associate professor.

Linenko Uliana Demyanovna – e-mail: uliana555555@mail.ru; phone: uliana555555@mail.ru; 333, Chekhov street, ap. 170, Taganrog, Russia; phone: +79043459313; student.

Pustovarova Aleksandra Vitalevna – e-mail: 000shurik000@rambler.ru; 22, Chekhov street, ap. 409, Taganrog, Russia; phone: +79885898232; student.

УДК 658.512

С.А. Ховансков, К.Е. Румянцев, В.С. Хованскова

МЕТОД ПОВЫШЕНИЯ ЗАЩИТЫ РАБОТОСПОСОБНОСТИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В КОМПЬЮТЕРНЫХ СЕТЯХ

Рассматривается подход к решению проблемы организации распределенных вычислений для выполнения объемных задач. Разработан алгоритм организации децентрализованных распределенных вычислений в вычислительной среде для решения сложных задач. Определена конфигурация системы распределенных вычислений с использованием локальной вычислительной сети на основе персональных компьютеров. Отсутствие центра управления системой и способность к адаптации при изменении параметров сети позволяет обеспечить системе высокую степень защиты ее работоспособности. На основе разработанного алгоритма предлагается решение задачи трассировки соединений методом организации распределенных вычислений.

Распределенные вычисления; многопроцессорная система.

S.A. Khovanskov, KE. Rumyantsev, V.S. Khovanskova

METHOD OF IMPROVING THE PROTECTION OF EFFICIENCY OF DISTRIBUTED COMPUTING IN COMPUTER NETWORKS

The article proposes an approach to the solution of the problem of organizing the distribution-specific calculations to perform the bulk of tasks. Developed the algorithm of bodies-organisation of decentralized distributed computing in computational environment for the solution of complex problems. Determined by the configuration of distributed computing systems with the use of local computer network on the basis of individually-owned computers. Absence of the control center of the system and the ability to adapt to change the settings of the network allows the system you are a high degree of protection it is running. On the basis of the developed algorithms were developed for-rhythm of a solution of the trace compounds method of organization of the distributed computing.

Distributed computing; multiprocessor system.

В настоящее время использование и внедрение вычислительных систем (ВС) стало одним из показателей научно-технического прогресса. Постоянно разрабатываются новые, более совершенные ВС. Несмотря на возрастание производительности ВС, происходит более быстрый рост сложности задач требующих решения при соблюдении жестких временных ограничений. Поэтому разработка методов повышения быстродействия ВС и эффективности их использования при решении задач является актуальной задачей.

Одним из направлений повышения быстродействия ВС и соблюдения требуемых временных критериев решения задач является разработка методов организации распределенных вычислений, а направление развития и использования систем на основе автономных вычислительных узлов концентрирует все необходимые для таких вычислений возможности с наибольшей выразительностью и полнотой [1].

Разработанные методы организации распределенных вычислений имеют такие недостатки, как:

- ◆ проблема разделения задачи на вычислительные модули (ВМ);
- ◆ проблемы стоимости (спецвычислители);
- ◆ наличие центров управления;
- ◆ жесткое назначение каждому центру ВМ задачи;
- ◆ низкая степень «живучести» в нестабильной вычислительной среде.

Поиск решения этих проблем привел к разработке метода реализации распределенных вычислений на основе мультиагентной системы, реализующей принцип коллективного принятия решения.

При организации распределенных вычислений каждому ВМ ориентированного графа задачи ставится в соответствие компьютер с учетом наличия физических связей в сети, способных реализовать потоки передаваемой информации между ВМ [1, 2]. Каждый компьютер, имеющий одного из агентов, работает независимо от других аналогичных компьютеров.

Используя мультиагентный подход, был предложен метод организации распределенных вычислений для решения задач из области автоматизированного проектирования, из которых основными по временным затратам являются задачи трассировки.

Разработан алгоритм на основе предлагаемого метода, выполняемый путем организации распределенных вычислений на нескольких взаимодействующих вычислительных узлах (ВУ) [2].

С точки зрения массовости и доступности узлов в качестве ВУ для организации распределенных вычислений была выбрана компьютерная сеть.

Для трассировки соединений выбран алгоритм параллельного построения связывающих деревьев цепей на коммутационном рабочем поле [3]. Его достоинствами являются простота, разрешение возникающих в ходе трассировки конфликтных и тупиковых ситуаций.

Все контакты цепи разбиваются на пары таким образом, чтобы при соединении этих пар контактов образовывалось дерево, включающее в себя все контакты.

Определим количество связываемых пар контактов S в множестве контактов M .

$$S = M - 1.$$

Соответственно и число связей для построения дерева будет равно количеству пар контактов S .

Связи образуются путем присвоения последовательности дискретов метки цепи, начиная с исходного контакта. Из исходного контакта метка передается одному из четырех соседних дискретов. Какому именно дискрету, определяется согласно последовательности приоритетных направлений. Этот дискрет включается в строящуюся цепь. Далее построение связи на дискретном рабочем поле между двумя дискретами, соответствующими начальному и конечному контактам, производится путем аналогичной последовательной передачи и присвоением свободным дискретам метки цепи (рис. 1, 2).

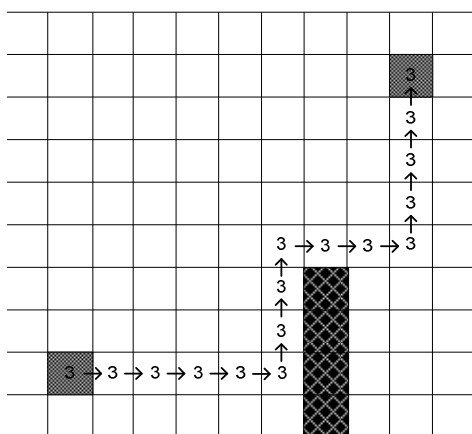


Рис. 1. Передача метки

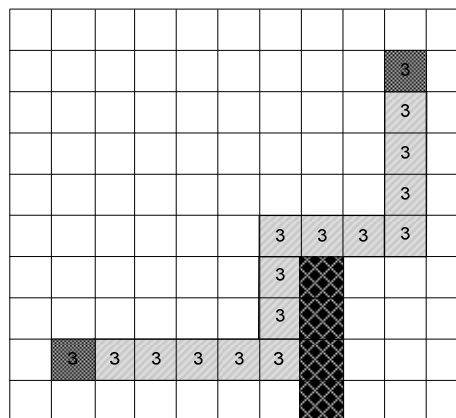


Рис. 2. Соединение контактов трассой

Алгоритм параллельного построения связывающих деревьев на коммутационном рабочем поле реализуется на МВС, имеющей решетчатую структуру и центральное устройство управления (ЦУУ). Множество узловых процессоров (УП), M_p образуют поле процессоров (ПП). Каждый УП $P(i, j) \in M_p$ может обмениваться информацией только с четырьмя соседними УП, определяемыми решетчатой структурой связей между ними.

Дерево строится на множестве дискретов D – рабочем поле коммутационного слоя. Рабочее поле D моделируется на поле УП следующим образом. Каждому дискрету рабочего поля с координатами x, y ставится в соответствие УП $P(i, j) \in M_p$, где i, j – координаты УП в ПП. Каждый дискрет $d(x, y) \in D$ имеет координаты x, y . Обозначим M_{ij} множество процессоров, соседних процессору $P(i, j)$, тогда дискрету $d(x, y) \in D$ соответствует процессор $P(i, j) \in M_p$.

Множество D_{xy} соседних дискретов для $d(x, y)$ составит:

$$D_{xy} = \{d(x+1, y), d(x, y+1), d(x-1, y), d(x, y-1)\}.$$

Таким образом, между элементами M_{ij} и D_{xy} существует взаимно-однозначное соответствие.

Обозначим A – множество всех цепей $A = \{A1, A2, \dots, Al, \dots, AL\}$, где L – число цепей схемы. Для каждой цепи необходимо построить ортогональное связывающее дерево G_l . Контакты, которые входят в одну цепь Al , имеют метку l . Ребрами, образующими связывающее дерево, являются ортогональные связи, соединяющие пары контактов цепи.

Из пары соединяемых контактов один выбирается в качестве "источника", второй – в качестве "цели", соответствующие им УП, помечаются аналогично. Из УП "источника" будет начинаться построение связи в соответствии с выбранными приоритетными направлениями. Задача состоит в построении связи из "источника" по направлению к "цели".

Построение связи заканчивается при достижении меткой, передаваемой через процессорное поле, узлового процессора с координатами "цели".

Для реализации алгоритма в каждый УП ЦУУ МВС засылает информацию о состоянии процессора и алгоритм его работы. Алгоритм, определяющий работу любого из УП, образующих процессорное поле, состоит из пяти этапов.

Перед началом построения во все $P(i, j) \in M_p$ ЦУУ МВС засылает информацию о состоянии соответствующего ему дискрета $d(x, y)$, поскольку алгоритм построения определяет работу каждого УП в зависимости от его состояния.

Обозначим S_{ij} – текущее состояние УП $P(i, j)$.

В соответствии с состоянием дискрета состояние процессора $P(i, j)$ кодируем следующим образом:

$$Y = \begin{cases} 0 & \text{– свободный УП;} \\ 1 & \text{– запрещенный;} \\ L & \text{– УП содержит часть связи цепи AL или контакт } k_{l\varphi}. \end{cases}$$

Кроме того, в каждый $P(i, j) \in M_p [S_{ij} = l; k_{l\varphi} \in K^*]$ засылается информация о координатах "цели" – $O_{l\varphi}^*$, соответствующих контакту.

Построение связи $Z(l, m)$ заключается во включении на каждом шаге построения в последовательность процессоров, имеющих метку l , очередного процессора $P(i, j) [S_{ij} = 0]$ изменением его состояния на $S_{ij} = l$.

Для нахождения оптимального решения задачи построения дерева введем ограничение – деревья цепей должны иметь минимальное количество пересечений друг с другом. Это позволит более равномерно распределить связи по рабочему полю.

Связи строятся на поле УП МВС одновременно из всех источников за счет независимости работы каждого УП. Построение связи ведется путем передачи метки цепи из P_i соседнему процессору P_{i+1} и присвоение ему метки P_i .

Для минимизации общей длины связи будем производить построение дополнительных контактов в точках слияния связей дерева.

Процесс построения каждой связи $Z(l,m)$ осуществляется до включения в последовательность процессоров, помеченных меткой l процессора $P(i,j)$, соответствующего контакту $k_{i\varphi} \in K^*$ или $P(i,j)[S_{ij} = l]$, через который уже прошла связь $Z(l,f) \in Z_l$, причем, координаты цели связи $Z(l,m)$ должны быть равны координатам источника связи $Z(l,f)$, а в месте слияния связей образуется новый контакт.

При решении задачи трассировки путем организации распределенных вычислений возникают трудности. Это связано с необходимостью обеспечения обменов между вычислительными узлами, а также с тем, что в отличие от МВС, полная автономность каждого вычислительного узла сети не позволяет использовать дискретное рабочее поле как единое целое.

Конфигурация системы представляет собой граф компьютерной сети с информационными потоками между узлами, имеющий решетчатую структуру.

Поскольку процесс трассировки выполняется на нескольких вычислительных узлах, то перед началом процесса все множество дискретов D разбивается на соответствующее количество частей.

На этапе разделения дискретов D между соседними частями определяются граничные линии, состоящие из множества дискретов d_i (рис. 3).

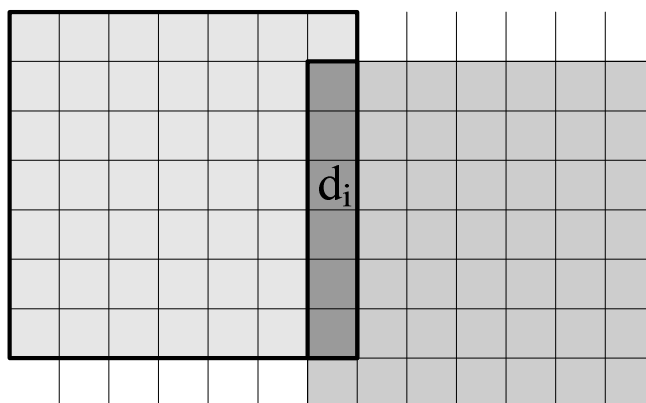


Рис. 3. Множество дискретов d_i , общих для двух узлов

При выполнении построения связывающих деревьев цепей каждый вычислительный узел работает независимо от других, он последовательно связь за связью реализует процесс построения цепей в своей части дискретного рабочего поля.

Для снижения степени влияния автономности узлов сети на конечный результат алгоритм построения был изменен для учета фактора взаиморасположения соединяемых контактов.

Если контакты («источники» и «цели») расположены в разных частях рабочего поля, то в этом случае выполняются построения связи до границы части рабочего поля вычислительного узла m .

Для продолжения этой трассы в соседней части рабочего поля процесс изменения списков трассируемых и протрассированных связей повторяется. Благодаря единому алгоритму построения связей и организации обмена информацией между узлами, рабочее поле для трассировки всех связей будет представляться как единое целое.

Для разрешения тупиковых ситуаций разработан алгоритм работы узлов.

1. Выбор первого приоритетного направления. Перейти к п. 2°.
2. Дискрет $a_i, j+1$ свободен? Если да, то к п. 8°, иначе к п. 3°.
3. Выбор второго приоритетного направления. Перейти к п. 4°.
4. Дискрет $a_i, j+1$ свободен? Если да, то к п. 8°, иначе к п. 5°.
5. Выбор второго приоритетного направления. Перейти к п. 6°.
6. Дискрет a_i+1, j свободен? Если да, то к п. 8°, иначе к п. 7°.
7. Дискрет a_i, j помечается как занятый и метка передается в дискрет $a_i, j-1$. Перейти к п. 9°.
- 8° Новый дискрет помечается меткой. Перейти к п. 9°.
- 9° Конец алгоритма.

Если трассировка соединения выполнена, то она вычеркивается из списка трассируемых соединений. Процесс трассировки прекращается, когда список трассируемых соединений текущего узла становится пустым.

Достоинством такого подхода, несомненно, является то, что такой метод организация распределенной вычислительной системы позволяет реализовать выполнение сложных задач на любом количестве узлов компьютерной сети. При этом нет определенных требований к рабочим узлам по производительности, по параметрам каналов связи. Необходимо только, чтобы объем памяти каждого из компьютеров позволял хранить всю исходную для задачи информацию. Количество узлов, составляющих распределенную вычислительную систему, не ограничено, но чем больше задействовано узлов в процессе трассировки, тем меньше время выполнения задачи. Отсутствие центра управления системой и способность к адаптации при изменении вычислительной среды позволяет обеспечить системе высокую степень живучести.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ховансков С.А., Литвиненко В.А. Решения задач путем организации распределенных вычислений в сети // Известия ЮФУ. Технические науки. – 2008. – № 3 (80). – С. 16-21.
2. Ховансков С.А., Литвиненко В.А., Норкин О.Р. Оптимизации распределенных вычислений на базе алгоритма реконfigurирования и продукции предметной области. Труды конгресса по интеллектуальным системам и информационным технологиям “AIS-IT’09”. – М.: Физматлит, 2009. Т. 2. – С. 153-158.
3. Ховансков С.А., Литвиненко В.А., Калашиников В.А. Алгоритм трассировки на многопроцессорной вычислительной системе. – Депонировано в ВИНТИ 19.04.91г., N1684-B91.

Статью рекомендовал к опубликованию д.т.н., профессор В.И. Марчук.

Хованскова Вера Сергеевна – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: sah59@mail.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634676616; студентка.

Ховансков Сергей Андреевич – тел.: 88634642530; кафедра ИБТКС; к.т.н.; доцент.

Khovanskova Vera Sergeevna – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: sah59@mail.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634676616; student.

Khovanskov Sergey Andreevich – phone: +78634642530; the department of ISTCN; cand. of eng sc. associate professor.