

Pomazanov Alexander Vasilievich – SouthernFederalUniversity; e-mail: pav_tsure@mail.ru; 2, Shevchenko street, Taganrog, 347922, Russia; phone: +78634312482; the department of telecommunication systems information security; associate professor; cand. of eng. sc.

Shibaev Stanislav Sergeevich – e-mail: sheab@list.ru; laboratory of nanophotonics and optoelectronics; senior scientist; cand. of eng. sc.

Volik Denis Petrovich – e-mail: DVLbest@mail.ru; the department of radioengineering electronics; assistant.

УДК 681.3.069, 681.324

С.А. Ховансков, О.Р. Норкин, С.С. Парфенова, В.С. Хованскова

АЛГОРИТМ ПОСТРОЕНИЯ СВЯЗЕЙ НА СЕТЕВЫХ КОМПЬЮТЕРАХ ПУТЕМ ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ*

Статья посвящена вопросу сокращения времени решения задачи построения связывающих деревьев цепей на дискретном рабочем поле. Предлагается алгоритм построения связывающих деревьев ориентированный на использование распределенных вычислений. При построении связей выполняется передача метки связи из помеченного дискрета поля в соседний свободный. Связь считается построенной, если непрерывная последовательность помеченных дискретов соединила два контакта. В качестве вычислительной среды была выбрана неоднородная компьютерная сеть. Алгоритм разрабатывался так, чтобы он выполнялся на нескольких сетевых вычислительных узлах. При его выполнении каждый вычислительный узел может работать независимо друг от друга, и процесс построения всех связывающих деревьев цепей на коммутационном рабочем поле выполняется на всех вычислительных узлах одновременно. При этом процесс выполнения алгоритма использует особенности взаимодействия вычислительных узлов в сети и достоинства организации децентрализованных распределенных вычислений. Для организации децентрализованных распределенных вычислений было предложено использовать технологию мобильных агентов. Проведенная оценка сокращения времени построения связывающих деревьев цепей при использовании мобильных агентов показала, что эффективность применения распределенной вычислительной системы, организованной по разработанному алгоритму повышается с увеличением количества строящихся цепей. Однако для решения задачи с заданными параметрами использование больше определенного количества компьютеров приводит к снижению эффективности распределенной системы. Как показал анализ эффективности применения распределенной вычислительной системы, основной причиной снижения показателя сокращения времени решения, при дальнейшей увеличении количества используемых вычислительных узлов является увеличение количества обменов между ними.

Распределенные вычисления; трассировка; вычислительный процесс; организация вычислений.

S.A. Khovanskov, O.R. Norkin, S.S. Parfenova, V.S. Khovanskova

THE ALGORITHM FOR CONSTRUCTING LINKS COMPUTERS ON A NETWORK THROUGH DISTRIBUTED COMPUTING

The article is devoted to solving the problem of reducing the time of construction of the spanning tree on the discrete circuits working field. The algorithm for constructing the spanning tree-oriented use of distributed computing is offered. Transfer label marked increment of the communication field in the adjacent free runs when links are building. Communication is considered to be built, if a continuous sequence of discrete labeled joined two contacts. Heterogeneous computer network has been selected as the computing environment. The algorithm is designed for

* Работа выполнена на основе гос. задания Минобрнауки РФ № 213.01-11/2014-9.

executing on multiple networked computing nodes. When it is running each compute node can operate independently of each other, and the process of construction of the spanning tree circuits on the switching operation is performed on the field all computing nodes simultaneously. The process of the algorithm uses the features of the interaction of computing nodes in the network and the dignity of the organization of decentralized distributed computing. To organize decentralized distributed computing was proposed to use the technology of mobile agents. The assessment to reduce the time of construction of the spanning tree circuits using mobile agents showed that the effectiveness of a distributed computing system, organized by developed algorithm increases with the number of circuits being built. However, to solve the problem with the specified parameters to use more than a certain number of computers reduces the efficiency of the distributed system. As the analysis of the effectiveness of a distributed computing system, the main reason for reducing the index to reduce the time solution, with further increase in the number of computing nodes is used by increasing the number of exchanges between them.

Distributed computing; tracing; computational process algorithms.

Введение. Очень часто при решении задач, требующих выполнения большого объема вычислений, одним из главных требований является решение задачи за минимальное время. Одним из путей обеспечения этого требования является – организация распределенных вычислений на вычислительной системе (ВС). Действия процесса решения задачи распределяются между несколькими центрами обработки данных [1–5, 10, 14]. В качестве центров обработки используются различные вычислительные узлы, а для выполнения задачи требуется организация совместной работы вычислительных узлов системы.

Традиционно распределенные вычисления решаются с помощью специальной многоагентной системы, состоящей из нескольких вычислительных узлов и управляемой специальной программой работающей на выделенном компьютере. С помощью управляющей программы за каждым компьютером закрепляется вычислительный блок задачи, организуется контроль работоспособности вычислительных узлов и процесса решения задачи, обеспечивается обмен и сбор полученных данных [1, 3, 4, 10]. При ее создании учитываются особенности решаемой задачи. Неизбежно возникает проблема разделения задачи на вычислительные блоки, распределение их между центрами обработки и синхронизации работы этих центров.

Из-за привязки управляющей программы к конкретной решаемой задаче быстрая перенастройка системы при изменении алгоритма вычислений просто невозможна. Также недостатками реализации такого подхода являются неизбежные временные потери, большие затраты на разработку программы и низкая живучесть системы из-за наличия единого центра управления, низкая эффективность решения задач с высокой степенью связности по данным.

Алгоритм организации распределённых вычислений. В настоящее время активно развивается направление, связанное с использованием технологии мобильных агентов [4, 5, 9, 15–19]. Достоинствами технологии мобильных агентов являются высокая надежность за счет децентрализации управления работой вычислительных узлов и возможность организации независимых вычислительных процессов [1–7].

Такая технология позволяет организовать систему распределённых вычислений на основе компьютерной сети, используя в качестве вычислительных узлов типовые персональные компьютеры (ПК).

Это может быть и обычная компьютерная сеть, построенная по топологиям, имеющим иерархическую структуру, использующим различные сетевые технологии (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 к компьютеру.

В данном контексте под агентами понимается программный модуль, работающий на каждом узле и управляющий всеми выполняемыми им вычислительными процессами. Организация распределенных вычислений для решения конкретной задачи выполняется путем процесса коллективного принятия решения мультиагентной системой. При этом с помощью мультиагентной системы возможно решение проблемы распределения решаемых блоков задачи в компьютерной среде.

Состав модуля мультиагентной системы (агент) включает в себя вычислительные модули необходимые для выполнения самой задачи обработки данных и управляющую программу агента, реализующую распределенные вычисления.

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 предоставляет пользователю результаты вычислений.

Все агенты являются копиями одного агента и работают по одному и тому же алгоритму. Агенты равноправны и управляют работой своего ПК независимо друг от друга. Под управлением своего агента ПК занимается сбором и обработкой информации. По окончании выполнения заданного объема вычислений выполняется обмен между агентами ПК имеющими связи согласно иерархической структуре. Первоначально модуль агента может располагаться на одном или нескольких ПК. Агенты готовы сразу при получении задачи приступить к ее выполнению.

Среди наиболее трудоемких задач решаемых системами автоматизированного проектирования можно выделить задачи разработки новых узлов ВС, из которых основными по временным затратам являются задачи трассировки. На основе технологии мобильных агентов был разработан алгоритм, позволяющий сократить время выполнения задачи трассировки. Компьютерная сеть используется для выполнения распределенных вычислений и получения обрабатываемой информации (входные данные) из одного источника во все ПК (например, сеть Интернет, решение задач САПР, задачи управления и др.). Все результаты обработок передаются по сети и собираются на одном из ПК.

На первом этапе организации вычисления весь вычислительный процесс задачи делят на несколько, связанных между собой. Задача представляется в виде графа B с M вершинами, где вершины, соответствуют программным блокам задачи, а ребра – информационные потоки между ними. Структура графа B представляет собой совокупность связанных вершин $(b_1, b_2, b_3, \dots, b_m)$. Эти вершины могут образовывать структуру из нескольких уровней.

Компьютеры с мобильными агентами соединяются в компьютерную сеть и образуют граф сети T с N вершинами, в котором вершины графа – соответствующие ПК сети. Работа в сети предполагает использование разделяемой среды передачи данных. Каждый компьютер может обмениваться с любым другим, поэтому обычно модель сети представляется в виде полного графа.

Реализация распределенных вычислений начинается с наложения графа B на граф T на рис. 1.

Для этой реализации должно быть выполнено условие $N > M$. В каждую вершину t_j записывается и активируется программный код агента a_i . Каждый агент содержит вычислительный блок b_i информацию о направлении потоков данных между соединенных с ним вершинами графа B . Наложение выполняется случайным образом, как только будут обнаружены в количестве M центры обработки информации.

Для организации распределенных вычислений в компьютерной сети каждой вершине b_i графа задачи B ставится в соответствие вершина t_i графа компьютерной сети T . Связи между компьютерами устанавливаются так, что они повторяют связи между соответствующими вершинами графа задачи B . При этом соответствие вершин графа задачи вершинам графа компьютерной сети, может носить случайный характер, поскольку программные модули графа B могут быть выполнены любым вычислительным узлом сети. Такая конфигурация вычислительной системы позволяет решать задачу, но время ее решения очень сильно зависит от полученного варианта наложения вершин графов задачи на граф сети, выполненного без учета производительности вершин графа сети и вычислительной сложности вершин графа задачи.

В соответствии с связями графа задачи B между ПК сети в процессе решения устанавливаются связи. Эта среда используется для получения данных, передачи результатов обработанных данных, обмена информацией между агентами.

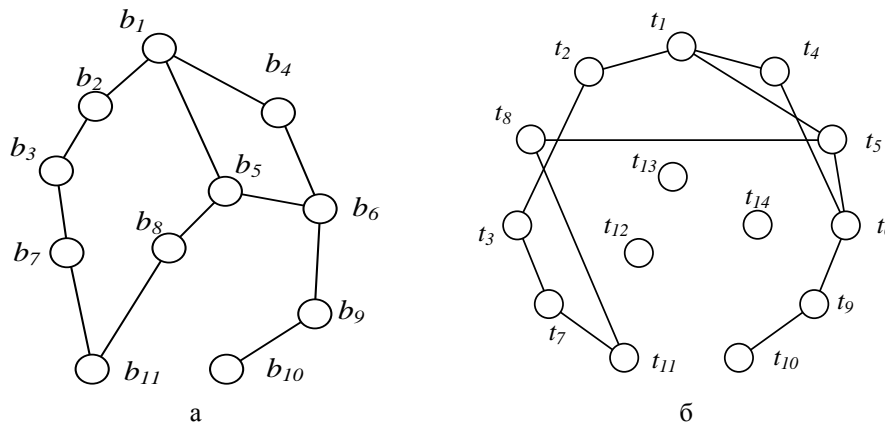


Рис. 1. Исходный граф B (а) и вариант наложения графов B и T (б)

Использование алгоритма организации распределённых вычислений для решения задач. Существует ряд алгоритмов ориентированных на использования распределенной вычислительной системы для ускорения процесса выполнения задачи трассировки. Например, алгоритм параллельного построения связывающих деревьев цепей на коммутационном рабочем поле [6–13, 20]. Этот алгоритм предназначен для выполнения трассировки соединений на многомашинной системе. Его достоинствами являются простота, разрешение возникающих в ходе трассировки конфликтных и тупиковых ситуаций, а также учет расположения проведенных соединений. Алгоритм построения связывающих деревьев цепей на коммутационном рабочем поле реализуется каждым вычислительным узлом, при этом каждый узел работает независимо от остальных и трассировка всех соединений выполняется одновременно.

На основе этого алгоритма разработан алгоритм параллельного построения связывающих деревьев цепей, использующий особенности взаимодействия вычислительных узлов в сети и достоинства организации распределенных вычислений [6, 9].

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

Процесс трассировки с использованием мультиагентной системы выполняется сразу на нескольких вычислительных узлах сети. Перед началом трассировки задача делится на вычислительные блоки, которые будут выполняться одновременно на ПК сети.

Для определения направления трассируемых соединений был разработан следующий алгоритм определения направления построения связей.

При ортогональном построении связи возможны четыре направления построения: вправо, влево, вверх и вниз. Для выполнения трассировки на компьютере информация о направлениях трассируемых соединений была закодирована согласно рис. 2.

Исходной информацией для работы этого пункта алгоритма является взаимоположение координат пары контактов "источника" $O_{i\phi}^+$ и "цели" $O_{i\phi}^*$, которые используются для определения номера варианта последовательности приоритетных направлений, по которым осуществляется построение.

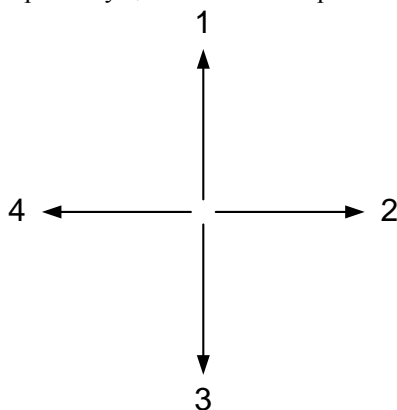


Рис. 2. Направления передачи информации

В соответствии с каждым из возможных соотношений координат "источника" и "цели" существует последовательность приоритетных направлений для построения связи. В табл. 1 приведены последовательности направлений возможных вариантов построения (рис. 2, 3).

Таблица 1

| номер варианта | Последовательность приоритетных направлений |
|----------------|---|
| 1 | 2,3,1,4 |
| 2 | 1,4,2,3 |
| 3 | 3,2,4,1 |
| 4 | 4,1,3,2 |
| 5 | 4,3,1,2 |
| 6 | 1,2,4,3 |
| 7 | 3,4,2,1 |
| 8 | 2,1,3,4 |

Варианты последовательностей приоритетных направлений хранятся в ОП каждого вычислительного узла (ВУ).

Вариант последовательности приоритетных направлений для построения связи определяется по разности координат "цели" $O_{l\varphi}^*$ и "источника". Пусть "цель" – дискрет $P(q,z)$, а "источника" – дискрет $P(i,j)$. Тогда вычисляем Δx и Δy :

$$\Delta x = i - q; \Delta y = j - z,$$

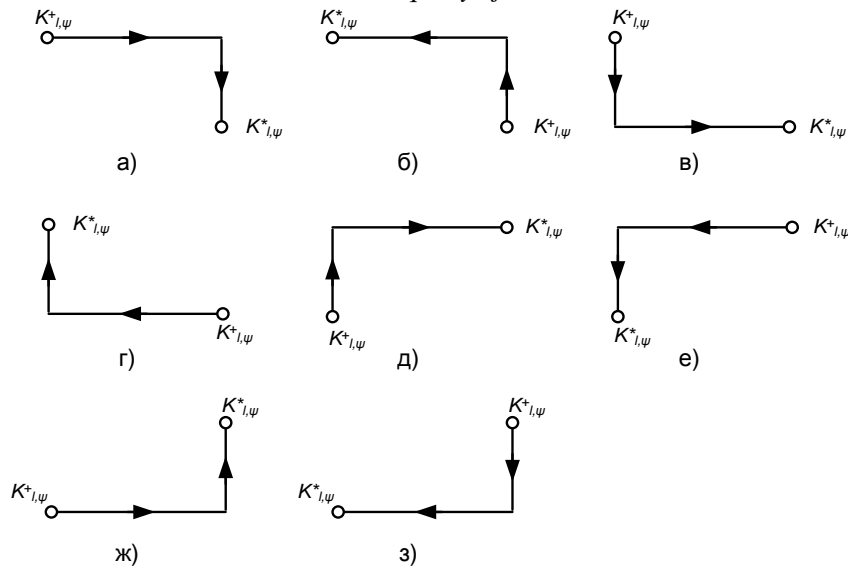


Рис. 3. Варианты реализации связей и взаиморасположение контактов:
 а) вариант 1-й; б) вариант 2-й; в) вариант 3-й; г) вариант 4-й; д) вариант 5-й;
 е) вариант 6-й; ж) вариант 7-й; з) вариант 8-й

Таблица 2

| Δx | Δy | Варианты |
|------------|------------|----------|
| + | - | 1 или 3 |
| - | + | 2 или 4 |
| - | - | 5 или 7 |
| + | + | 6 или 8 |

В зависимости от знака Δx и Δy выбираем любой один из двух вариантов приоритетных направлений, в соответствии с табл. 2.

По последовательности приоритетных направлений производится поиск свободного дискрета в множестве M_{ij} и выполняется построение связи $Z(l,m)$ с меткой l .

По последовательности приоритетных направлений производится поиск свободного дискрета в множестве M_{ij} и выполняется построение связи $Z(l,m)$ с меткой l .

Выбранный вариант последовательности приоритетных направлений является информацией о связи и передается на третьем этапе.

Рассмотрим алгоритм, по которому обрабатывается информация для свободных дискрет. Под свободным дискретом мы понимаем дискрет $P(i,j)$, через который может быть проложена связь $Z(l,m)$, т.е. $P(i,j) \in M_p [S_{ij}=0]$.

Исходной информацией для работы каждого ВУ является информация о состояниях соседних дискрет и информация о связи, полученная на предыдущем шаге. Обозначим W_{qz} направление передачи информации от дискрета $P(i,j)$ к дискрету $P(q,z)$. Сформулируем алгоритм построения связи.

Алгоритм 1

1. Если координата i дискрета $P(i,j)$ совпала с x_{lf}^* , то переход к п.3, иначе переход к п.2.
2. Если координата j дискрета $P(i,j)$ совпала с y_{lf}^* , то переход к п.3, иначе переход к п.4.
3. Заменить вариант приоритетных направлений.
4. Выбрать очередной дискрет $P(q,z) \in M_{ij}$ согласно варианту приоритетных направлений, переход к п.5.
5. Если состояние дискрета $S_{qz} = 0$, то переход к п.8, иначе переход к п.6.
6. Если состояние дискрета $S_{qz} > 1$, то переход к п.7, иначе переход к п.9.
7. Если дискрет $P(q,z) \in M_{ij} [S_{qz} = 1]$, то переход к п.13, иначе к п.9.
8. Запомнить направление W_{qz} .
9. Если все дискреты множества M_{ij} , просмотрены, то переход к п.10, иначе переход к п.4.
10. Если $W_{qz} \neq 0$, то переход к п.14, иначе переход к п.11.
11. Зарегистрировать блокировку связи $Z(l,m)$ и переход к п.12.
12. Выбрать направление к дискрету $P(e,f) [S_{ef} = 1]$, из которого строилась связь через дискрет $P(i,j)$.
13. Запомнить направление W_{qz} .
14. Конец работы алгоритма.

Для дискрета $P(i,j) \in M_{ij} [S_{qz} \neq 0]$ при выборе направления используется исходная информация. Просмотр состояний S_{qz} дискрета $P(q,z) \in M_{ij}$ производится в последовательности, соответствующей номеру варианта приоритетных направлений. Для соблюдения такой последовательности реально просмотр осуществляется в трех направлениях, поскольку на четвертом направлении находится предпоследний дискрет строящейся связи $Z(l,m)$, и производится в такой очередности: третье направление в последовательности, второе, первое. Таким образом, остается выбранным наилучшее свободное направление.

После анализа информации $P(i,j)$ и $P(q,z)$ на третьем этапе предыдущего шага, для дискрета $P(i,j)$ на втором этапе следующего шага сравнивается его состояние с состоянием дискрета $P(q,z)$. Если состояния $P(q,z)$ не изменилось, то $P(i,j)$ продолжает оставаться последним дискретом строящейся связи и надо искать новое направление для ее построения.

Поскольку направления распространения связей ортогональны, то для соединения двух контактов с разными координатами необходимо хотя бы один раз изменить направление построения связи, что осуществляется заменой варианта приоритетных направлений. С этой целью для дискрета $P(i,j)$ сравниваются координаты i и j с координатами O_{lf}^* связи $Z(l,m)$.

При несовпадении ни по одной из координат вариант приоритетных направлений передачи метки остается без изменений.

Если выполняется одно из условий $i = x_{lf}^*$ или $j = y_{lf}^*$, т.е. дискрет $P(i,j)$ находится на одной из координат с "целью", то в этом случае вариант приоритетных направлений, использованный на предыдущем шаге, заменяется на другой вариант. Следующая передача информации о связи $Z(l,m)$ от дискрета $P(i,j)$ осуществляется по направлению, выбранному согласно новому варианту последовательности. Замена вариантов приоритетных направлений на новые осуществляется в соответствии с табл. 3.

В зависимости от состояний соседних дискрет для дискрета $P(i,j)$ принимается решение об изменении направления построения связи.

Если $P(q,z) \in M_{ij}$ имеет состояние $S_{qz} > 1$, то $P(i,j)$ сравнивает состояние S_{qz} с меткой l связи $Z(l,m)$. В том случае, если $S_{qz} \neq S_{ij}$, то выбирается направление к первому $P(q,z) \in M_{ij}$, удовлетворяющего условию $S_{qz} = 0$, для передачи в $P(q,z)$ информации о связи на п.3 алгоритма.

Таблица 3

| | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|
| старый вариант | 2 | 3 | 6 | 7 | I | 4 | 5 | 8 |
| новый вариант | 5 | 8 | I | 4 | 7 | 6 | 3 | 2 |

Если существует хотя бы один дискрет $P(q,z) \in M_{ij}, [S_{ij} = S_{qz}]$, то передача метки от дискрета $P(i,j)$ в $P(q,z)$ осуществляется на третьем этапе. При передаче используется следующая информация:

- ◆ координаты "цели" (конечного контакта) $O_{l\phi}^*$, к которому строится связь $Z(l,m)$;
- ◆ координаты "источника" (начального контакта) $O_{l\phi}^+$;
- ◆ номер варианта последовательности приоритетных направлений для связи $Z(l,m)$;
- ◆ метка l связи $Z(l,m)$.

Оценка ускорения выполнения алгоритма построения связей. Оценим ускорение выполнение процесса построения всех связей L при изменении количества компьютеров K соединенных в сеть и используемых для реализации распределенных вычислений.

В общем виде время решения задачи равно

$$T_p = t_d * d_i + t_{перед}, \quad (1)$$

где t_d – время увеличения длины связи на один дискрет; d_i – суммарное количество дискретов во всех связях строящихся на i -том компьютере; $t_{перед}$ – время обмена информацией между соседними компьютерами.

Для простоты будем считать, что производительность компьютеров в сети приблизительно одинакова, поэтому время увеличения длины связи на один дискрет будет одинаковым для всех компьютеров сети.

Время передачи определяется количеством передач p_i выполняемых при построении связей i -м компьютером

$$t_{перед} = p_i * t_{обм}. \quad (2)$$

Само число p_i равно количеству дискретов рабочего поля, которые являются общими у пары соседних компьютеров. Поскольку части дискретного рабочего поля у всех компьютеров примерно одинаковы, то можно записать

$$p_i = 2 * (a_i + b_i), \quad (3)$$

где a_i и b_i размеры дискретного поля i -го компьютера.

Если размеры общего рабочего дискретного поля записать как $A*B$, то

$$a_i = \frac{A}{\sqrt{K}}; b_i = \frac{B}{\sqrt{K}} \quad (4)$$

Подставляя значения в формулу (1), найдем суммарное количество дискретов во всех связях строящихся на i -м компьютере

$$d_i = \frac{\left(\frac{A+B}{\sqrt{K}} + 4\right) \cdot L}{2 \cdot K}. \quad (5)$$

Тогда время решения задачи построения T_p будет равно

$$T_p = \frac{\left(\frac{A+B}{\sqrt{K}} + 4\right) \cdot L}{2 \cdot K} \cdot \left(\frac{A+B}{\sqrt{K}}\right) \cdot 2 \cdot t_d \cdot t_{обм}. \quad (6)$$

Используя полученную формулу (6) построим графики зависимости общего времени решения задачи построения связей от количества задействованных сетевых компьютеров (рис. 4, 5), используемых для организации распределенных вычислений.

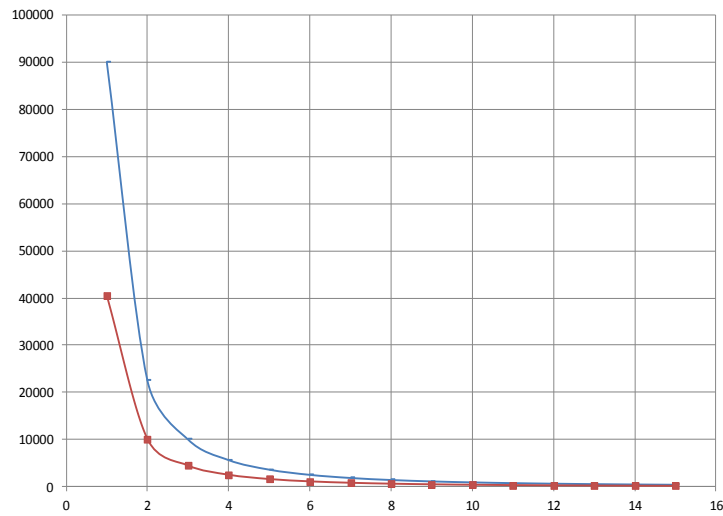


Рис. 4. Зависимость времени решения задачи от количества компьютеров для $L=1000$ и $L=600$ при A и B равные 100 и 200 дискртков соответственно

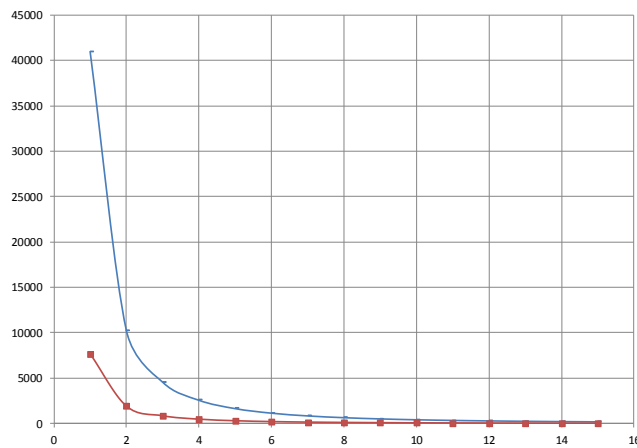


Рис. 5. Зависимость времени решения задачи от количества компьютеров для $L=4000$ и $L=1000$ при A и B равные 200 и 3000 дискртков соответственно

Заключение. Предлагаемый алгоритм на распределенной вычислительной системе позволяет выполнить трассировку соединений любой длины. Для организации распределенных вычислений не требуется единого управляющего центра. Благодаря одновременной реализации процесса построения связей несколькими вычислительными узлами, сокращается общее время трассировки. Разработанный алгоритм позволяет разрешать ситуации блокирования трассируемых соединений переходящих из одной части дискретного рабочего поля вычислительного узла в соседнюю часть другого узла. Обмен между узлами производится минимальным объемом информации.

Достоинством такого подхода, несомненно, является то, что такая организация распределенной вычислительной системы позволяет реализовать выполнение трассировки 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 трассировки информацию и программу трассировки. Количество узлов составляющих распределенную вычислительную систему не ограничено, но чем больше задействовано узлов в процессе трассировки, тем меньше время выполнения задачи.

Оценка ускорения выполнение процесса построения всех связей L при изменении количества компьютеров с помощью математической модели показала, что при увеличении количества компьютеров общее время решения задачи построения связей по разработанному алгоритму уменьшается. Эффективность использования распределенной вычислительной системы, организованной по разработанному алгоритму, повышается с увеличением количества строящихся трасс. Однако для решения задачи с заданными параметрами использование больше определенного количества компьютеров приводит к снижению эффективности распределенной системы. Основной причиной снижения показателя сокращения времени решения при дальнейшей увеличении количества используемых компьютеров является увеличение количества обменов между ними.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. 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.
2. Ховансков С.А., Литвиненко В.А. Оптимизация решения задачи в распределенных системах // Известия ТРТУ. – 2005. – № 3 (47). – С. 209-210.
3. Ховансков С.А., Литвиненко В.А. Решения задач путем организации распределенных вычислений в сети // Известия ЮФУ. Технические науки. – 2008. – № 3 (80). – С. 16-21.
4. Ховансков С.А., Румянцев К.Е., Хованскова В.С. Алгоритм работы мультиагентной системы для ускорения решения задач моделирования путем организации распределенных вычислений // Материалы XIII Международной научно-практической конференции «ИБ-2013». Ч. II. – Таганрог: Изд-во ЮФУ, 2013. – С. 16-22.
5. Ховансков С.А., Румянцев К.Е., Хованскова В.С. Алгоритм программного модуля децентрализованной системы управления для создания безопасных распределенных вычислений в неустойчивой вычислительной среде // Известия ЮФУ. Технические науки. – 2013. – № 12 (149). – С. 247-253.
6. Ховансков С.А., Литвиненко В.А., Калашиников В.А. Алгоритм трассировки на многопроцессорной вычислительной системе. – Депонировано в ВИНТИ 19.04.91 г., № 1684-В91.
7. Ховансков С.А. Распараллеливание алгоритмов построения связывающего дерева для решения на многопроцессорной вычислительной системе // Известия ТРТУ. – 1997. – № 3 (6). – С. 226.

8. Литвиненко В.А., Ховансков С.А., Рябов О.В., Платонов В.А. Разработка модуля размещения учебно-исследовательской САПР на базе САПР Ki-CAD // Известия ЮФУ. Технические науки. – 2008. – № 4 (81). – С. 68-73.
9. Ховансков С.А., Литвиненко В.А., Норкин О.Р. Организация распределенных вычислений для решения задач трассировки // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 48-55.
10. Litvinenko V.A., Khovanskov S.A., Norkin O.R., Khovanskova V.S. The algorithm for determining the direction of building relations in a distributed computing system // IS&IT'12: proceedings of the Congress on intelligent systems and information technologies. Scientific publication in 4 vol. Vol. 4. – Moscow: Physmathlit, 2012. – P. 49.
11. Ховансков С.А., Литвиненко В.А., Литвиненко Е.В. Модифицированный адаптивный алгоритм определения максимальных полных подграфов // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 227-231.
12. Ховансков С.А., Литвиненко В.А., Литвиненко Е.В. Гибридный метод управления точностью решения экстремальных задач на графах // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 112-116.
13. Ховансков С.А., Литвиненко В.А., Максютя Д.Ю. Адаптивный алгоритм построения дерева Штейнера // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 152-160.
14. Defining the Cloud Computing Framework // Cloud Computing Journal. <http://cloudcomputing.sys-con.com/node/612375/>.
15. Ховансков С.А., Норкин О.Р. Алгоритм повышения производительности распределенных сетевых вычислений // Информатизация и связь. – 2011. – № 3. – С. 96-98.
16. Madkour A.M., Eassa F.E., Ali A.M., Qayyum N.U. Mobile-Agent-Based Systems Against Malicious Hosts // World Applied Sciences Journal. – 2014. – Vol. 29 (2). – P. 287-297.
17. Muñoz A., Pablo A., Maña A. Multiagent Systems Protection // Advances in Software Engineering. – 2011. – Article ID 281517. – 9 p. – doi:10.1155/2011/281517.
18. Xudong G., Yiling Ya., Yinyuan Y. POM-a mobile model against malicious hosts // Proceedings of High Performance Computing in the Asia-Pacific Region. – 2000. – Vol. 2. – P. 1165-1166.
19. Ховансков С.А., Мельник Э.В., Блушвили И.В. Метод организации распределенных вычислений в управляющих системах // Мехатроника, автоматизация, управление. – 2003. – № 4. – С. 9-12.
20. Ховансков С.А., Загурский М.Ю., Хованскова В.С., Литвиненко В.А. Алгоритм трассировки связывающих деревьев, предназначенный для выполнения на распределенной вычислительной системе // Информатика, вычислительная техника и инженерное образование. – 2014. – № 5 (20). – С. 23-28.

REFERENCES

1. Khovanskov S.A., Litvinenko V.A., Norkin O.R. Optimizatsii raspredelennykh vychisleniy na baze algoritma rekonfigurirovaniya i produktsiy predmetnoy oblasti [Optimization of distributed computing based on the reconfiguration algorithm and products subject area], *Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam "AIS-IT'09"* [Proceedings of the Congress on intellectual systems and information technologies "AIS-IT'09"]. Moscow: Fizmatlit, 2009, Vol. 2, pp. 153-158.
2. Khovanskov S.A., Litvinenko V.A. Optimizatsiya resheniya zadachi v raspredelennykh sistemakh [Optimization of decision tasks in distributed systems], *Izvestiya TRTU* [Izvestiya TSURE], 2005, No. 3 (47), pp. 209-210.
3. Khovanskov S.A., Litvinenko V.A. Resheniya zadach putem organizatsii raspredelennykh vychisleniy v seti [The organization of fulfilling tasks by the method of a cooperative decisions making], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 3 (80), pp. 16-21.
4. Khovanskov S.A., Rumyantsev K.E., Khovanskova V.S. Algoritm raboty mul'tiagentnoy sistemy dlya uskoreniya resheniya zadach modelirovaniya putem organizatsii raspredelennykh vychisleniy [The algorithm of multi-agent system to accelerate the decision of problems of modeling by distributed computing], *Materialy XIII Mezhdunarodnoy nauchno-prakticheskoy konferentsii «IB-2013»* [Proceedings of the XIII International scientific-practical conference "Information security 2013"]. Part II. Taganrog: Izd-vo YuFU, 2013, pp. 16-22.

5. *Khovanskov S.A., Rumyantsev K.E., Khovanskova V.S.* Algoritm programmogo modulya detsentralizovannoy sistemy upravleniya dlya sozdaniya bezopasnykh raspredelennykh vychisleniy v neustoychivoy vychislitel'noy srede [The algorithm of the program module of a decentralized management system to create a secure distributed computing in an unstable environment], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 12 (149), pp. 247-253.
6. *Khovanskov S.A., Litvinenko V.A., Kalashnikov V.A.* Algoritm trassirovki na mnogoprotsessornoy vychislitel'noy sisteme [Trace the algorithm on a multiprocessor computer system]. Deposited in VINITI 19.04.91, No. 1684-V91.
7. *Khovanskov C.A.* Rasparallelvanie algoritmov postroeniya svyazyvayushchego dereva dlya resheniya na mnogoprotsessornoy vychislitel'noy sisteme [Parallelization of algorithms for constructing connecting the tree to a multiprocessor computer system], *Izvestiya TRTU* [Izvestiya TSURE], 1997, No. 3 (6), pp. 226.
8. *Litvinenko V.A., Khovanskov S.A., Ryabov O.V., Platonov V.A.* Razrabotka modulya razmeshcheniya uchebno-issledovatel'skoy SAPR na baze SAPR Ki-CAD [The resulting matrix of interactive geometrical transformations composition search method in CAD], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 4 (81), pp. 68-73.
9. *Khovanskov S.A., Litvinenko V.A., Norkin O.R.* Organizatsiya raspredelennykh vychisleniy dlya resheniya zadach trassirovki [The organization of the distributed calculations for the decision of problems of trace], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12 (113), pp. 48-55.
10. *Litvinenko V.A., Khovanskov S.A., Norkin O.R., Khovanskova V.S.* The algorithm for determining the direction of building relations in a distributed computing system, *IS&IT'12: proceedings of the Congress on intelligent systems and information technologies. Scientific publication in 4 vol.* Vol. 4. Moscow: Phymathlit, 2012, pp. 49.
11. *Khovanskov S.A., Litvinenko V.A., Litvinenko E.V.* Modifitsirovanny adaptivnyy algoritm opredeleniya maksimal'nykh polnykh podgrafov [The modified adaptive algorithm of definition of the maximum full subgraphs], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 227-231.
12. *Khovanskov S.A., Litvinenko V.A., Litvinenko E.V.* Gibridnyy metod upravleniya tochnost'yu resheniya ekstremal'nykh zadach na grafakh [Hybrid method by accuracy of the decision of extreme tasks on graphs], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 112-116.
13. *Khovanskov S.A., Litvinenko V.A., Maksyuta D.Yu.* Adaptivnyy algoritm postroeniya dereva Shteynera [Adaptive algorithm of the Steiner's tree creation], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 152-160.
14. Defining the Cloud Computing Framework, *Cloud Computing Journal*. Available at: <http://cloudcomputing.sys-con.com/node/612375/>.
15. *Khovanskov S.A., Norkin O.R.* Algoritm povysheniya proizvoditel'nosti raspredelennykh setevykh vychisleniy [Algorithm to improve the performance of distributed network computing], *Informatizatsiya i svyaz'* [Informatization and communication], 2011, No. 3, pp. 96-98.
16. *Madkour A.M., Eassa F.E., Ali A.M., Qayyum N.U.* Mobile-Agent-Based Systems Against Malicious Hosts, *World Applied Sciences Journal*, 2014, Vol. 29 (2), pp. 287-297.
17. *Muñoz A., Pablo A., Maña A* Multiagent Systems Protection, *Advances in Software Engineering*, 2011. Article ID 281517, 9 p. doi:10.1155/2011/281517.
18. *Xudong G., Yiling Ya., Yinyuan Y.* POM-a mobile model against malicious hosts, *Proceedings of High Performance Computing in the Asia-Pacific Region*, 2000, Vol. 2, pp. 1165-1166.
19. *Khovanskov S.A., Mel'nik E.V., Bluishvili I.V.* Metod organizatsii raspredelennykh vychisleniy v upravlyayushchikh sistemakh [A method of organizing distributed computations in control systems], *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, Automation, Control], 2003, No. 4, pp. 9-12.

20. *Khovanskov S.A., Zagurskiy M.Yu., Khovanskova V.S., Litvinenko V.A.* Algoritmi trassi-rovki svyazyvayushchikh derev'ev, prednaznachennyu dlya vypolneniya na raspredelennoy vychislitel'noy sisteme [The algorithm for trace linking trees, designed to run on a distributed computing system], *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Information, Computing and Engineering Education], 2014, No. 5 (20), pp. 23-28.

Статью рекомендовал к опубликованию д.т.н. С.Г. Капустян.

Ховансков Сергей Андреевич – Южный федеральный университет; e-mail: sah59@mail.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 8634371902; кафедра информационной безопасности телекоммуникационных систем; к.т.н.; доцент.

Хованскова Вера Сергеевна – e-mail: v.s.khovanskova@gmail.com; кафедра информационной безопасности телекоммуникационных систем, аспирантка.

Норкин Олег Рауфатович – e-mail: oran@tgn.sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 8634371787; кафедра системного анализа и телекоммуникаций; к.т.н.; доцент.

Парфенова Светлана Саркисовна – e-mail: svetlanapar66@mail.ru; кафедра системного анализа и телекоммуникаций; ассистент.

Khovanskov Sergey Andreevich – Southern Federal University; e-mail: sah59@mail.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634371902.; the department of information security of telecommunication systems; cand. of eng. sc.; associate professor.

Khovanskova Vera Sergeevna – e-mail: v.s.khovanskova@gmail.com; the department of information security of telecommunication systems; postgraduate student.

Norkin Oleg Raufatovich – e-mail: oran@tgn.sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371787; the department of system analysis and telecommunications; cand. of eng. sc.; associate professor.

Parfenova Svetlana Sapkisovna – e-mail: svetlanapar66@mail.ru; the department of system analysis and telecommunications; assistant.

УДК 614.841.3, 614.842

Е.В. Воробьев, А.Я. Номерчук, К.К. Арутюнов, Г.Г. Лисовой, В.И. Плотников **ТЕХНИЧЕСКИЕ КОМПОНЕНТЫ ДЛЯ СОЗДАНИЯ СИСТЕМЫ РАННЕГО** **ОБНАРУЖЕНИЯ ЛЕСНЫХ ПОЖАРОВ**

В связи с глобальным потеплением, а также расширением деятельности человека проблема лесных пожаров в настоящее время стоит чрезвычайно остро. Для своевременного обнаружения лесных пожаров используются системы, включающие ряд технических средств, алгоритмы их работы, аналитические модели, а также модели для предсказания возникновения пожаров как на основе данных о местности, погодных характеристиках, так и на основе данных с сенсорных устройств. Целью данной статьи является освещение технических компонентов, которые с различной долей успеха могут быть использованы для обнаружения лесных пожаров. Основным способом решения максимального раннего, точного и эффективного детектирования очагов возгорания является объединение различных методов и приборов детектирования в единую систему. Данные должны быть рассмотрены в полуавтоматическом режиме в рамках единой математической модели, адаптированной для данного региона, типа леса, климатических условий. Основными элементами такой системы целесообразно считать: систему датчиков различного уровня, распределенных в окружающей среде с учетом рельефа местности и вероятности образования очагов возгорания; радиолокационные устройства, калиброванные для обнаружения дыма; тепловизоры и камеры; подсистема передачи данных, построенная на протоколах и оборудовании с низким электропотреблением; серверное устройство, получающее и обрабатывающее информацию с участием человека. Система, предлагае-