

УДК 004.032.24

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

**АЛГОРИТМ ПРОГРАММНОГО МОДУЛЯ ДЕЦЕНТРАЛИЗОВАННОЙ
СИСТЕМЫ УПРАВЛЕНИЯ ДЛЯ СОЗДАНИЯ БЕЗОПАСНЫХ
РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В НЕУСТОЙЧИВОЙ
ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ**

Существует множество сложных задач имеющих жесткое ограничение по времени решения. С целью сокращения времени решения таких задач и повышения безопасности вычислительного процесса предлагается метод организации распределенных вычислений на базе масштабируемой компьютерной сети. Сеть представляет собой неустойчивую вычислительную среду, состоящую из автономных узлов. Для организации вычислений в сети используется мультиагентная система. Под мультиагентной системой понимается совокупность одинаковых программных модулей расположенных на компьютерах сети. В ходе выполнения задачи все агенты, с целью сокращения времени решения, образуют децентрализованную систему управления распределенными вычислениями. Эта система позволяет реализовать такие функции как динамическая оптимизация назначений вычислительных блоков задач компьютерам в соответствии с их вычислительными ресурсами, слежение за процессами выполнения задач и в случае необходимости переназначение вычислительной нагрузки между узлами. Это позволяет обеспечить безопасность и «живучесть» создаваемых вычислительных процессов в компьютерной сети.

Распределенные вычисления; многовариантное моделирование; мультиагентная система; оптимизация конфигурации вычислительной системы.

S.A. Khovanskov, C.E. Rumyantsev, V.S. Khovanskova

**THE ALGORITHM OF THE PROGRAM MODULE OF A DECENTRALIZED
MANAGEMENT SYSTEM TO CREATE A SECURE DISTRIBUTED
COMPUTING IN AN UNSTABLE ENVIRONMENT**

There are many complex tasks with strict time limit solutions. To reduce the time for solving such problems and improve the security of computing process we propose a method of organizing distributed computing based on the scalable network. The Network is an unstable computing environment consisting of autonomous nodes. For the organization of computing network uses multi-agent system. Under the multi-agent system is a set of identical software modules located on network computers. During task all the agents in order to reduce solution time, form a decentralized system of management of distributed computing. This system allows for features such as dynamic optimization purposes of computing units computer tasks according to their computing resources, tracking progress of the task and, if necessary, reassign the processing load between nodes. This allows you to ensure the safety and "persistence" generated computing processes in a computer network.

Distributed computing using multivariate modeling; multiagent system; optimize the configuration of a computer system.

В настоящее время множество задач требуют выполнения большого объема вычислений за минимальное время. К ним относятся задачи моделирования.

Сегодня задачи моделирования, решают не только математики и инженеры, создающие сложные системы безопасности или дорогостоящие технические сооружения. Необходимость их решения возникает и у многих политических деятелей, дипломатов, администраторов и людей иных профессий, которым требуется принимать решения на основе прогнозирования развития текущих процессов часто с выбором наилучшего варианта решения задачи.

Решая задачи моделирования, мы имеем дело с многомерными динамическими задачами, где все величины зависят от времени: что создаёт определенные сложности и трудности. При этом, несмотря на множество возможных решений должен быть выбран только один вариант. Но какой именно вариант или какую комбинацию вариантов в данный момент и в данной ситуации выберет если ситуация острая и от этого выбора зависит многое.

На практике задачи моделирования требуют выбора наилучших решений не по одному, а сразу по нескольким критериям (многокритериальные задачи оптимизации), которые приносят дополнительные и не всегда разрешимые трудности. Если объект и его параметры являются переменными, зависящими от времени, то в этом случае используются более сложные стратегии решения задач моделирования.

Перед многокритериальными задачами логико-математические методы падают. Здесь число оптимальных решений равно числу критериев оптимальности (для каждого критерия своё), а понятие общего оптимума теряет смысл. Многокритериальность создаёт ситуацию неопределённости. Для выбора конечного решения существует аппарат прикладной математики, приспособленный для решения с помощью вычислительных машин многих практических задач оптимизации. Несмотря на это, проблема поиска оптимальных решений за приемлемое время становится все более актуальной [1, 2].

Прежде всего, выбор одного из многих вариантов может потребовать огромного перебора параметров, недоступного даже для самой быстродействующей ЭВМ. Подсчитано, например, что при решении задачи распределения 20 критериев по 10 объектам число возможных вариантов составит 10^8 . Даже если расчет каждого варианта потребует всего 10 арифметических операций, то и тогда общее число расчетных операций достигнет миллиарда, что не может быть выполнено ЭВМ в приемлемые сроки.

С другой стороны, чем больше выбор альтернативных вариантов, тем больше вероятность отыскания наилучшего решения. Многовариантные задачи можно рассмотреть за приемлемое время лишь с помощью логико-математических методов оптимизации и ускорения выполнения задачи на современных компьютерах.

Самым популярным решением этой проблемы в настоящее время является использование распределенных вычислений [1–5].

В качестве вычислительной среды для организации распределенных вычислений может использоваться либо мультипроцессорная вычислительная машина, либо многомашинная вычислительная система, либо обычная компьютерная сеть.

Из них наиболее доступной вычислительной системой, на которой возможно выполнять распределенные вычисления, является сеть, обладающая достаточным или избыточным количеством центров обработки данных (локальная, глобальная сети). В ней при организации распределенных вычислений каждому вычислительному блоку задачи должен ставиться в соответствие компьютер с учетом пропускной способности физических связей в сети, способных реализовать потоки передаваемой информации между вычислительными блоками [2, 3, 6].

При организации выполнения распределенных вычислений в обычной компьютерной сети основными проблемами являются организация совместной работы компьютеров в сети, распределение между ними вычислительной нагрузки, поддержка работоспособности системы, подбор компьютеров в сети.

Методов организации такой распределенной вычислительной системы множество. Основным недостатком этих методов является использование труда программиста на конфигурацию и настройку системы под конкретную задачу [2, 3], а также расход вычислительных ресурсов и времени на оптимизацию системы, способной удовлетворять требованиям к отводимым временным ресурсам [4, 6].

Наряду с этими методами существует множество специальных языков программирования, которые позволяют программистам организовывать распределенные вычисления в сети для решения объемных задач. Недостатком такой организации являются значительные временные затраты. Это накладывает определенные ограничения на быстрдействие решения задачи и предъявляет требования к используемым вычислительным ресурсам как по надежности, так и по производительности. В основном таким путем организуется решение задач, имеющих между вычислительными блоками низкую степень связности по данным.

Основной задачей решаемой авторами было создание метода позволяющего свести к минимуму подготовительные этапы для решения любой задачи, использовать в качестве вычислительных центров любые компьютеры сети с нестабильной вычислительной средой, оптимизировать время выполнения задачи за счет подбора компьютера обладающего достаточными вычислительными ресурсами для выполнения конкретного вычислительного блока. В тоже время создаваемая система должна быть работоспособной при любом наборе компьютеров, как по количеству, так и по производительности, иметь высокую степень безопасности не терять работоспособность и выполнять решение за отведенные под задачу временные ресурсы при динамическом изменении набора компьютеров составляющих масштабируемую вычислительную сеть [4, 6].

В качестве наиболее перспективного пути организации выполнения распределенных вычислений в компьютерной сети было выбрано использование мультиагентной системы реализующей метод коллективного принятия решения [6].

Под мультиагентной системой в данном случае понимается набор агентов, каждый из которых представляет программный модуль и помещается на отдельном компьютере. Агент выполняет управление только своим компьютером и поэтому его работа является независимой. Он организует решение задач на своем компьютере, инициирует обмен данными с компьютерами других агентов, выполняет обработку полученной от других агентов информации и на ее основе принимает решения.

Все агенты образуют однородную систему и работают по одному и тому же алгоритму. Отличительной особенностью мультиагентной системы является способность увеличивать число образующих ее компьютеров путем передачи копии программного модуля агента в свободные компьютеры. Благодаря тому, что агенты работают по одному алгоритму, возможно решение поставленных задач методом принятия коллективного решения.

Для решения перечисленных выше основных проблем был разработан алгоритм работы агента мультиагентной системы который позволяет организовать распределенную вычислительную систему на основе мультиагентной системы.

Основной алгоритм программы-агента реализует следующие задачи:

- ◆ получение информации от компьютеров-источников;
- ◆ выполнение на компьютере требуемых операций над полученной информацией;
- ◆ передача результата компьютеру-получателю.

Основная проблема, возникающая при такой организации распределенных вычислений, заключается в том, что компьютеры в сети обладают разными вычислительными мощностями. Это приводит к тому, что при организации распределенных вычислений возможны множество различных вариантов назначения вычислительных блоков задачи компьютерам сети, которые оказывают различное влияние на время выполнения задачи. Это пытаются учитывать, предварительно выбирая из множества наиболее оптимальный вариант, обеспечивающий требуемое время решения задачи [7, 8].

Существует несколько путей для решения проблемы выбора нужного варианта. Задача выбора может быть упрощена путем подбора для организации распределенных вычислений компьютеров, имеющих одинаковую вычислительную мощность. Недостатком этого пути является ограничение круга решаемых задач только задачами обрабатываемыми большой массив данных и состоящих из одного небольшого вычислительного блока. Другой путь заключается в предварительном поиске оптимального варианта назначения из множества имеющихся [4]. Основным недостатком этого пути является то, что выбор оптимального варианта назначения вычислительных блоков задачи компьютерам сети требует дополнительных временных затрат, которые могут значительно превышать время решения самой задачи.

Для организации распределенных вычислений все множество операций, выполняемых при решении задачи, предварительно делят по функциональным признакам на вычислительные блоки, образуя граф B с M количеством блоков, а вычислительная система, в данном случае сеть, представляется в виде графа вычислительных ресурсов T имеющего N центров обработки данных.

В [3, 5] предложен алгоритм выбора оптимального варианта обрабатываемый агентами. Реализация распределенных вычислений начинается с наложения графа B на граф T . Для этой реализации должно быть выполнено условие $N > M$. В каждую вершину t_j записывается и активируется программный код агента a_j . 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.

После запуска вычислительных процессов данные поступают входным агентам частями. По мере их обработки данные передаются в направлении выходных агентов в соответствии с связями графа B . Обработка агентом одной части данных является шагом.

После каждого шага агенты передают соседним агентам обработанные данные и обмениваются с ними служебной информацией. Получая служебную информацию, агенты на фоне решения задачи выполняют поиск методом парных перестановок оптимального варианта наложения вершин графа B на вершины графа T .

Алгоритм поиска основан на том, что среди всех маршрутов передачи данных от входных агентов к конечным всегда есть критический маршрут, по которому передача данных занимает наибольшее время. Это время и является определяющим фактором общего времени решения задачи. Агенты, делая парные перестановки, перебирают варианты наложения и находят вариант, уменьшающий время передачи данных по критическому маршруту, а в конечном итоге уменьшающий общее время обработки данных.

Во время решения задачи под управлением агентов выполняется оптимизация вычислительной системы для ускорения процесса решения. Каждому вычислительному блоку графа задачи наиболее оптимально ставится в соответствие вычислительный узел графа вычислительной системы. Алгоритм поиска продолжает обрабатываться агентами и после того как оптимальный вариант наложения вершин графа B на вершины графа T будет найден. Но при этом время решения задачи на данной конфигурации вычислительной системы уже невозможно уменьшать из-за постоянства имеющихся вычислительных ресурсов графа T .

Для примера на рис. 1,а изображен исходный граф B с количеством блоков $M=11$.

Для организации распределенных вычислений выбрана локальная сеть T , имеющая значение $N=14$, при этом выполняется условие $N > M$. Граф B был наложен случайным образом на граф T и после нескольких шагов работы вычислительной системы был найден оптимальный вариант наложения (рис. 1,б) графа B на граф T .

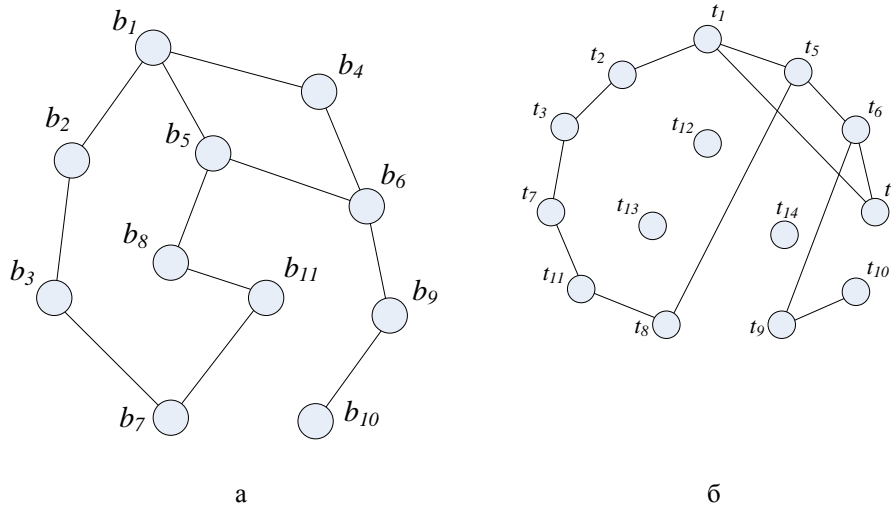


Рис. 1. Исходный граф B (а) и вариант наложения графов B и T (б)

С целью дальнейшего сокращения общего времени обработки данных разработан алгоритм работы агентов, позволяющий продолжить уменьшать общее время решения задачи.

Алгоритм агента

1. Получение входных данных.
2. Формирование адреса узла сети.
3. Этот адрес использовался? Да то к п.1.
4. Передача по сформированному адресу вербуемого узла служебной информации.
5. Пришел ответ от вербуемого узла? Нет, то к п.1.
6. Передача служебной информации соседним узлам графа B о завербованном узле.
7. Конец алгоритма.

Алгоритм основан на привлечении дополнительных вычислительных устройств в процесс решения некоторых вычислительных блоков графа B задачи. Поскольку вычислительная система представляет собой совокупность компьютеров объединенных в сеть или являющихся частью компьютерной сети. Это дает возможность вовлекать (вербовать) в процесс решения задачи «свободных» компьютеров, т.е. не вошедших в граф вычислительной структуры, что и позволяет продолжить сокращение времени решения задачи.

Свободные компьютеры могут быть завербованы программой-агентом находящейся на одном из компьютеров графа T .

Вербовать может любой агент. Вербовка заключается в том, что программа-агент передает свою копию на свободный компьютер и активизирует ее в случае если на нем нет программы-агента.

После нахождения оптимального варианта назначения общее время прохождения данных по вершинам графа задачи по-прежнему определяется критическим маршрутом.

Каждый агент определяет свое местонахождение относительно критического маршрута. Агент принадлежащий критическому маршруту и выполняющий b_i за определенное время вербует вычислительный узел $w \in (N - M)$.

Для этого он передает служебную информацию соседним с ним компьютерам и вычислительный модуль агенту завербованного компьютера.

Появляется новая вершина в графах задачи и вычислительной системы. Таким образом, может быть завербован случайным образом любой компьютер сети, например t_{14} .

После включения завербованного компьютера в вычислительную систему в постоянно повторяющемся процессе оптимизации варианта назначения будут участвовать и вычислительные ресурсы узла W . После нескольких шагов будет найдено новое оптимальное соответствие между вершинами графов задачи и вычислительной системы имеющее меньшее общее время прохождения данных по вершинам графа задачи.

В приведенном на рис. 1 варианте время прохождения данных через вычислительную систему при оптимальном варианте определяется критическим маршрутом $r_{крит} = (t_1, t_5, t_6, t_9, t_{10})$. После выполнения агентом узла t_9 алгоритма вербовки свободного узла был завербован узел t_{14} . После обмена между узлами t_9 и t_{14} информацией граф T изменился (рис. 2).

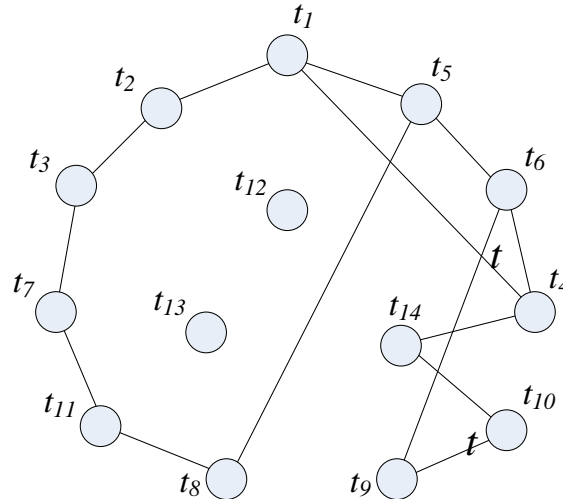


Рис. 2. Вариант наложения измененного графа B и графа T

Благодаря этому время прохождения данных через вычислительную систему уменьшилось, поскольку оно определяется новым критическим маршрутом $r_{крит} = (t_1, t_5, t_8, t_{11})$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Хованскова В.С., Румянцев К.Е., Ховансков С.А.* Метод повышения защиты работоспособности распределенных вычислений в компьютерных сетях // Известия ЮФУ. Технические науки. – 2012. – № 4 (129). – С. 102-107.
2. *Ховансков С.А., Норкин О.Р.* Алгоритм повышения производительности распределенных сетевых вычислений // Информатизация и связь. – 2011. – № 3. – С. 96-98.
3. *Литвиненко В.А., Ховансков С.А.* Решения задач путем организации распределенных вычислений в сети // Известия ЮФУ. Технические науки. – 2008. – № 3 (80). – С. 16-21.
4. *Ховансков С.А., Норкин О.Р.* Алгоритмическая оптимизация конфигурации системы децентрализованных вычислений // Телекоммуникации. – 2012. – № 11. – С. 2-5.
5. *Ховансков С.А.* Организация распределенных вычислений с иерархической структурой связей // Информационное противодействие угрозам терроризма. – 2007. – № 9. – С. 170-178.
6. *Литвиненко В.А., Ховансков С.А.* Алгоритм оптимизации параметров компьютерной сети для уменьшения времени решения задачи на основе мультиагентной системы // Известия ЮФУ. Технические науки. – 2007. – № 2 (77). – С. 186-190.
7. *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.
8. *Калашников В.А., Трунов И.Л., Ховансков С.А.* Параллельный алгоритм размещения на многопроцессорной вычислительной системе // Известия ЮФУ. Технические науки. – 1997. – № 3 (6). – С. 181-184.

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

Румянцев Константин Евгеньевич – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: rke2004@mail.ru; 347900, г. Таганрог, ул. Чехова, 2; тел.: 88634328725; кафедра ИБТКС; зав. кафедрой; д.т.н.; профессор,

Ховансков Сергей Андреевич – e-mail: sah59@mail.ru; тел.: 88634642530; кафедра ИБТКС; к.т.н.; доцент.

Хованскова Вера Сергеевна – e-mail: wepok@mail.ru; тел.: 88634676616; студентка.

Rumyantsev Constantine Evgen'evich – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: rke2004@mail.ru; 2, Chekhov street, Taganrog, 347900, Russia; phone: +78634328725; the department of IBTKS; head the department; dr. of eng. sc.; professor.

Khovanskov Sergey Andreevich – e-mail: sah59@mail.ru; phone: 88634642530; the department of ISTCN; cand. of eng. sc.; associate professor.

Khovanskova Vera Sergeevna – e-mail: wepok@mail.ru; phone: +78634676616; student.

УДК 004.056

Е.Н. Ефимов, Г.М. Лапицкая

**ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И БИЗНЕС-ПРОЦЕССЫ
КОМПАНИИ**

Цель исследования показать, что информационная безопасность должна создаваться и существовать в рамках достижения бизнес-целей компании. При этом бизнес-процессы являются первоисточником данных для организации информационной безопасности компании.

При разработке стратегии управления информационной безопасностью предложена комплексная диагностика состояния компании с помощью SWOT-анализа с условной количественной оценкой результатов, полученной с помощью метода анализа иерархий.