

*Скобелев Петр Олегович,
Граничин Олег Николаевич,
Будаев Денис Сергеевич,
Ларюхин Владимир Борисович,
Майоров Игорь Владимирович*

УДК 534

МУЛЬТИАГЕНТНАЯ СИСТЕМА ПЛАНИРОВАНИЯ ЗАДАЧ В ПРОГРАММНО- КОНФИГУРИРУЕМЫХ СЕТЯХ

Аннотация

Рассматривается мультиагентная система планирования задач в программно-конфигурируемых сетях. Система предназначена для моделирования распределения и выполнения задач на вычислительных ресурсах с учетом динамических характеристик и топологии сети.

Ключевые слова: мультиагентная система, планирование в реальном времени, адаптивность, программно-конфигурируемые сети.

1. ВВЕДЕНИЕ

Развитие компьютерных сетей до настоящего времени шло, в основном, экстенсивным путем: все большее число устройств объединялось в локальные и глобальные сети, совершенствовалось сетевое оборудование, увеличивалась пропускная способность каналов передачи данных, повышалось быстродействие серверов и сетевых компонентов.

Однако принципиально новые решения на практике, как правило, широко не внедрялись и постепенно существующие сети перестают справляться с возросшим количеством устройств и потоков данных. Сетевой трафик, его структура изменились в связи с

тем, что каждый запрос создает большое количество обращений к серверам приложений, базам данных, другим системам хранения и обработки информации. Сложность современной сетевой инфраструктуры и рост трафика приводят к росту отказов и при существующих тенденциях способны вызвать резкое ухудшение скоростных характеристик передачи данных.

При этом существующая сетевая архитектура по-прежнему ориентирована на группу достаточно старых протоколов для обеспечения взаимодействия усложняющихся сетевых устройств, что затрудняет масштабирование. Добавление новых устройств требует больших усилий, как и перенастройка коммутаторов, маршрутизаторов и сетевых экранов. Сетевая структура остается статичной, и поэтому необходимы значительные средства для её поддержки.

© Скобелев П.О., Граничин О.Н.,
Будаев Д.С., Ларюхин В.Б.,
Майоров И.В., 2013

Такие инновации, как облачные вычисления и виртуализация, требуют новых подходов к организации сетей.

В этой связи в последние несколько лет возрос интерес к новой архитектуре программно-конфигурируемых сетей или ПКС (SDN, Software Defined Networks), которые должны позволить преодолеть наметившийся кризис [1].

Концепция ПКС определяет смещение парадигмы в сетевой архитектуре, когда слой управления сетью программируется напрямую и отделяется от непосредственной маршрутизации пакетов. Такой перенос управления обеспечивает абстракцию основной сети для приложений верхних слоёв, что позволяет рассматривать сеть как логическую или виртуальную сущность. Разработанный протокол OpenFlow (OF), применяющийся теперь во многих сетевых устройствах, позволяет переместить функции управления маршрутизаторами сети в центральный модуль, называемый контроллером или сетевой операционной системой [2].

По сути, контроллер является промежуточным звеном, связывающим реальные сетевые устройства и некоторые приложения, реализующие логику работы оборудования. Протокол OpenFlow определяет саму возможность и рамки взаимодействия между контроллером и сетевым оборудованием. Функции пересылки пакетов остаются внутри маршрутизаторов, называемых форвардерами, или коммутаторами по той причине, что они не принимают решения о выборе маршрута, а лишь выполняют заданные контроллером правила маршрутизации. Контроллер (в совокупности с приложениями) можно считать сосредоточением интеллекта системы, поскольку решения о маршрутах принимаются именно на этом уровне, причем для потоков пакетов, а не пакетов по отдельности, после чего сетевые устройства конфигурируются через протокол OpenFlow. Протокол позволяет обеспечить полное представление сети на контроллере, мониторинг всех ресурсов и виртуализацию сети, что позволяет создавать сложные решения для управления сетью [3].

В этой связи задача является актуальной и значимой для моделирования и исследования различных вариантов организации сети.

2. ПОСТАНОВКА ЗАДАЧИ

Одной из активно развивающихся в настоящее время концепций является модель распределенных вычислений. Она основана на предоставлении сетевого доступа к общим вычислительным сетевым ресурсам, что значительно уменьшает расходы пользователей на создание соответствующей инфраструктуры и гибко реагирует на все изменения вычислительных потребностей. Распределенные вычисления, управляемые ПКС, могут быть более эффективны с точки зрения скорости работы, поскольку сеть может быть оптимизирована многочисленными специализированными алгоритмами, учитывающими топологию и динамические характеристики сети. Одной из задач оптимизации вычислений в распределенной системе сетевых устройств является задача балансировки нагрузки, активно исследуемая в настоящее время. В работе [4] рассматривается применение теории игр к задаче обеспечения необходимого уровня сервиса в системе конкурирующих за ресурсы приложений. Задача оптимизации качества обслуживания в распределенных динамических вычислительных системах [5] исследуется с помощью теории вероятностей. Предлагается рассчитывать локальные времена выполнения задач, длины очередей, времена ожиданий и другие метрики. Проведено моделирование доступности ресурсов в зависимости от алгоритмов планирования и характеристик вычислительной системы. Подход с использованием рандомизированных алгоритмов рассматривается в [6]. Задачи балансировки загрузки узлов вычислительной сети в стохастической постановке рассмотрены в [7].

В данной работе рассматривается мультиагентный подход [8, 9] к планированию вычислительных задач в распределенной системе на основе разрабатываемой модели сети агентов задач и ресурсов.

3. МЕТОД РЕШЕНИЯ И ОПИСАНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ ПЛАНИРОВАНИЯ

Имеется система, состоящая из нескольких вычислительных серверов, соединенных через коммутаторы OpenFlow в сеть, управляемую контроллером ПКС. Серверы предназначены для выполнения задач разного типа с различной скоростью, зависящей от типа задачи и от производительности ресурса (сервера). В систему поступает поток задач, каждая из которых может быть выполнена на ограниченном подмножестве серверов (рис. 1)

Соответствие серверов и типов задач определяется в описании каждой задачи. Задачи поступают в произвольные моменты времени и перераспределяются по системе. В результате на каждом сервере создается очередь задач на выполнение. Порядок перераспределения задач определяется в логическом слое контроллера. При этом учитываются динамические характеристики каналов передачи данных, состояния очередей в каждом устройстве сети (коммутаторах, серверах), соответствие типов задач и серверов. Контроллер ПКС динамически определяет наиболее подходящие сервера для выполне-

ния задач, учитывая, в том числе, и маршруты движения задач в системе. При необходимости (например, в случае непредвиденных отключений узлов сети) производится динамическое перепланирование. Общая цель моделирования – обеспечить выполнение задач за наименьшее время при сбалансированной загрузке всех узлов сети.

Динамическое распределение задач по серверам сети предлагается проводить с помощью мультиагентного подхода. Основным сущностям системы ставятся в соответствие программные агенты: агент задачи, агент ресурса (сервера), агент коммутатора. Агенты описываются атрибутами, включающими наиболее существенные поля, свойства и правила взаимодействия. Каждый агент имеет свою цель и стремится к ее достижению, отправляя и получая сообщения от других агентов. Для решения задач оптимизации состояний агентов вводятся «виртуальные стоимости», позволяющие применить рыночный подход в мультиагентной системе. При этом происходят изменения значений атрибутов агентов, поэтому в каждый момент времени существует некоторое состояние мультиагентной системы, отвечающее за локально оптимальное решение.

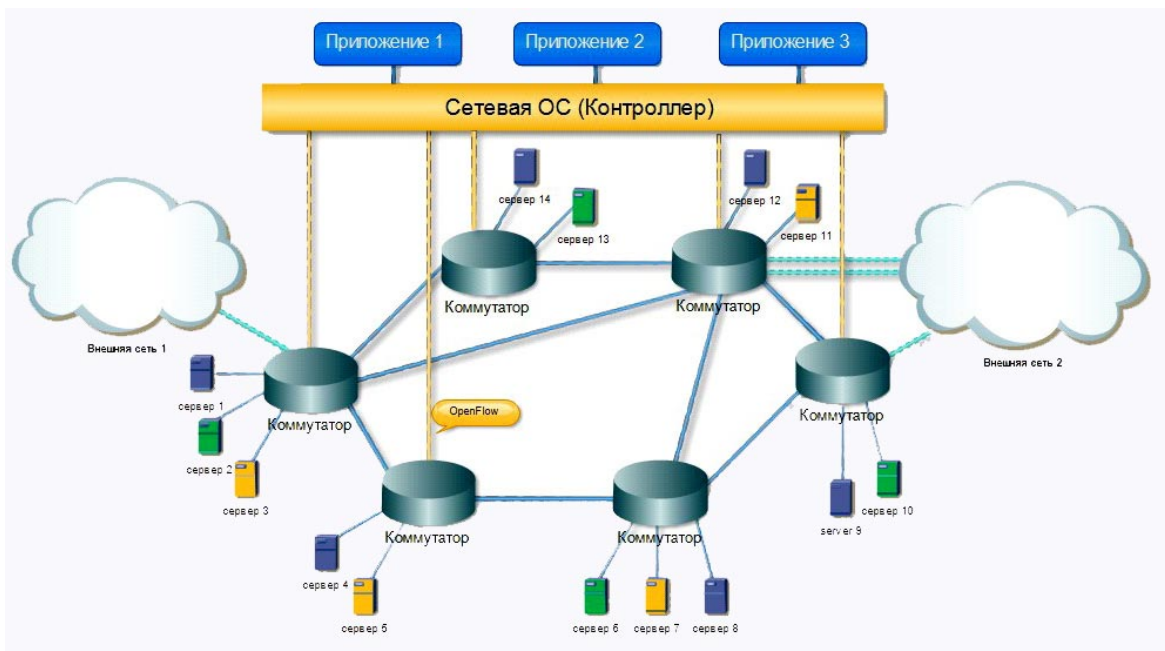


Рис. 1. Топология программно-конфигурируемой сети

4. ЛОГИЧЕСКАЯ АРХИТЕКТУРА СИСТЕМЫ

Система состоит из базы знаний – онтологии, в которой содержатся описания типов концептов: агентов задач, агентов ресурсов и агентов коммутаторов. Устанавливаются атрибуты, свойства и отношения между ними. Сцена виртуального мира служит для взаимодействия экземпляров агентов через обмен сообщениями. В сцене производится динамическое изменение значений атрибутов агентов в результате переговоров. Генератор внешних событий создает потоки входящих в систему задач с различными статистическими характеристиками, например, события изменения производительности и типа серверов, периоды их недоступности, подключения и отключения выделенных фрагментов сети.

Подсистема пользовательского интерфейса позволяет запускать моделирование событий, визуализировать граф сети, динамики стоимостей задач, распределения задач и очередей на серверах, загруженности каналов передачи и другие (рис. 2)

Агент задачи

Агент задачи представляет вычислительную задачу, которую нужно запланировать и выполнить с наименьшими затратами на одном из серверов сети ПКС.

В момент поступления каждой задачи в систему создается агент задачи, атрибутами которого в данной предметной области будут следующие поля:

- время поступления задачи,
- объем данных задачи,
- тип задачи (сопоставляется типу сервера),
- вычислительная сложность задачи, если известна,
- приоритет задачи (высокий, средний, низкий), если известен,
- желательное время начала задачи, если известно,
- желательное предельное время выполнения задачи, если известно,
- фактическое время начала выполнения задачи на сервере,
- фактическое время окончания выполнения задачи на сервере,
- состояние текущего виртуального счета.

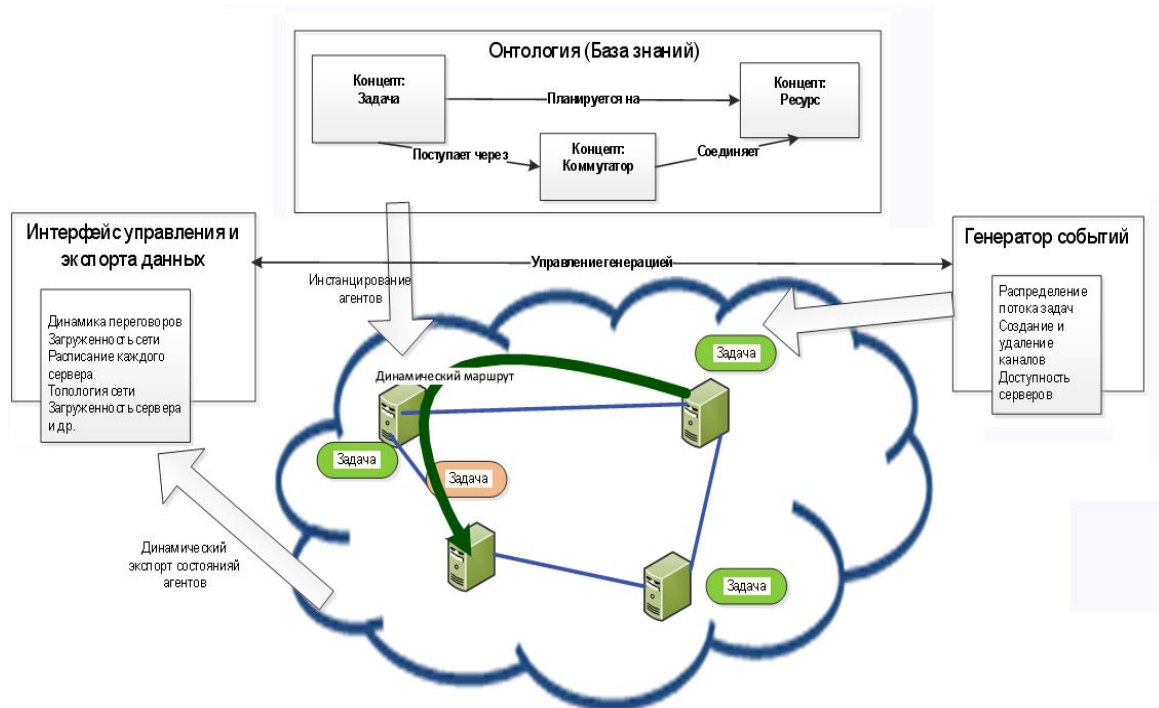


Рис. 2. Логическая архитектура системы

С каждым агентом задачи связывается кусочно-линейная функция бонусов/штрафов, которая изменяет текущий виртуальный счет агента задачи в зависимости от соотношения желательных и фактических времен, приоритета и вычислительной стоимости задачи. Вычислительная сложность и, соответственно, предельное время окончания представляют собой оценочные верхние границы, достоверно не известные до начала выполнения задачи на сервере. Фактические значения определяются в момент начала реального выполнения на основании некоторой статистики.

Агент ресурса (сервера)

Агент ресурса представляет в системе сервер, на котором могут выполняться задачи. Цель агента сервера – запланировать как можно больше задач и получить максимальную виртуальную прибыль от выполнения задач.

Атрибутами агента ресурса в данной предметной области будут следующие поля:

- набор типов задач, которые могут выполняться на данном сервере,
- производительность ресурса, определяющая расчетное время выполнения задач,
- текущий список задач, запланированных на ресурсе,
- допустимый объем очереди задач,
- таблица стоимости единицы времени ресурса,
- состояние текущего виртуального счета ресурса.

С каждым агентом ресурса ассоциируется штрафная функция, зависящая от количества запланированных задач на указанном горизонте планирования.

Агент коммутатора

Агенты коммутаторов предоставляют в сцену актуальную информацию о состоянии сети и призваны обеспечивать эффективное распределение задач по узлам дальнейшего следования.

Агент коммутатора описывается следующими полями:

- допустимый объем очереди задач,
- пропускная способность каждой связи с ближайшими агентами коммутаторов,

- текущий список задач для передачи по каждой связи агентам коммутаторов,
- состояние текущего виртуального счета коммутатора.

5. ФУНКЦИОНИРОВАНИЕ СИСТЕМЫ

Первоначально система имеет определенную топологию соединений, например, кольцо. Задачи поступают случайным образом на один или несколько коммутаторов из внешних сетей. При поступлении новой задачи система автоматически выполняет планирование в три стадии:

1. Стадия бесконфликтного планирования

Агент задачи ищет описания ресурсов в сцене, которые отвечают указанному типу, а также являются ближайшими с точки зрения времени доставки через имеющиеся на пути коммутаторы. Логическая сущность сцены мультиагентной системы по сути соответствует физической сущности контроллера ПКС. В результате работы со сценой агент задачи создает список из наиболее подходящих ресурсов. Далее агент формирует сообщение с просьбой об определении условий начального размещения задачи на ресурсе с учетом ее параметров, которое посылается всем агентам отобранных ресурсов. Агенты ресурсов анализируют свои расписания и рассчитывают возможность и стоимость размещения задачи, формируя встречные предложения со сдвинутыми сроками относительно запрошенных. В результате для всех отправленных сообщений агент ресурса получает набор возможных вариантов размещения задачи на ресурсы с указанием стоимости размещения, которая в дальнейшем должна быть списана с виртуального счета задачи и зачислена на счет выбранного ресурса. По имеющейся таблице вариантов агент задачи выбирает наилучший вариант по изначально заданным критериям с учетом их весов.

Если задача успешно разместилась на ресурс, это фиксируется «договором» (отношением) в сцене, и бюджет задачи обновляется, становясь равным стоимости работы за

вычетом затрат на ресурс. Агент ресурса также обновляет свой счет, увеличивая бюджет на указанную им стоимость, и вставляет работу в расписание. Определяются времена начала и окончания, опережения и задержки.

2. Стадия проактивности задач

Проактивность задач состоит в активации тех задач, которые уже запланированы, но еще не выполнены. Их положение в расписании ресурса может быть невыгодным с точки зрения агента задачи. Если агент задачи обладает достаточной суммой виртуальных денег, он может динамически попытаться перепланироваться на другой подходящий ресурс, компенсируя возможные ухудшения состояний агентов других задач в ходе разрешения конфликтов по положениям задач в расписании.

При наличии штрафных функций проактивность агентов задач – это процесс, управляемый разницей цен таблицы стоимостей ресурса, видом штрафных функций вовлеченных в конфликт агентов задач и зависящий от текущего виртуального бюджета работ. Стадия проактивности призвана сформировать обновленное равновесное состояние системы при количественном улучшении состояния задач.

3. Стадия проактивности ресурсов и определение маршрутов

Проактивность ресурса заключается в изменении плана размещения задач, которое начинается по инициативе ресурса. Ресурс при недостаточно плотном расписании или, наоборот, при слишком больших очередях при наличии событий поступления задач может снимать с планирования часть мало-выгодных для него задач. Маршруты движения задач по сети определяются с помощью выбора наименее загруженных каналов. При необходимости транспортировки выгодной задачи на ресурс создаются новые каналы, если имеется достаточный виртуальный бюджет. Динамические возникающие каналы, обеспечивающие локальную оптимальность маршрутизации, определяются функционированием логического виртуального слоя программно-конфигурируемой сети.

6. ПРЕИМУЩЕСТВА И ОГРАНИЧЕНИЯ

Преимущества предлагаемого мульти-агентного подхода заключаются в следующем.

- онтологическое описание сцены мира вычислительных ресурсов, задач и конфигурации сети позволяет вводить новые сущности ПКС и задавать их характеристики без значительных затрат на перепрограммирование;

- динамическое управление состоянием всех объектов сцены мира ПКС с помощью сценариев (скриптов);

- изменения в топологии сети могут производиться в режиме реального времени;

- динамическое планирование задач на серверы в мультиагентной системе, обеспечивающееся постоянной проактивностью задач и ресурсов на основе самоорганизации, основанной на стремлении агентов к максимизации виртуальной прибыли в актах взаимодействия путем обмена сообщениями;

- локальная оптимальность и непротиворечивость текущего расписания;

- локальная оптимальность путей маршрутизации пакетов и очередей задач;

- учет изменений в сетевом окружении и в статистических характеристиках потоков задач в реальном времени (адаптивность) не приводит к полному перерасчету расписаний, что уменьшает вычислительные затраты системы моделирования;

- динамическое перераспределение трафика с учетом загруженности каналов;

- простота логики взаимодействия агентов на основе максимизации виртуальной прибыли.

Разработанная система составляет платформу для моделирования и проведения исследований по выбору логики работы оборудования и программной системы управления сетью.

7. РЕАЛИЗАЦИЯ

Система разрабатывается как приложение для ОС Windows с использованием платформы .NET Framework на языке программирования C#. Ядро прототипа реализуется

в виде библиотеки (сборки), используя которую при необходимости можно расширить существующую логику моделирования. Разработка пользовательского интерфейса выполняется с применением технологии WPF. Для отображения графа сети используется библиотека Graph#.

Текущая реализация системы позволяет представить исследуемую сеть визуально в виде графа, производить начальную конфигурацию сети, а также определять параметры отдельных устройств: объемы для очередей задач отдельных коммутаторов и серверов, пропускные способности каналов связи, производительности серверов обработки, включены или выключены устройства (рис. 3).

В ходе процесса моделирования возможно изменение масштаба модельного времени относительно реального, что позволяет ускорять или замедлять процесс моделирования по желанию пользователя системы. Также в верхней части пользовательского интерфейса отображается лог событий системы. Каждое событие (приход новой задачи, выполнение задачи на ресурсе) фиксируется в логе и доступно для анализа в любой момент моделирования.

Система позволяет сохранять ранее сформированные конфигурации сети в файл. Но помимо этого система способна сохранять текущее состояние самого процесса моделирования в любой момент времени. При этом моделируемые устройства сохра-

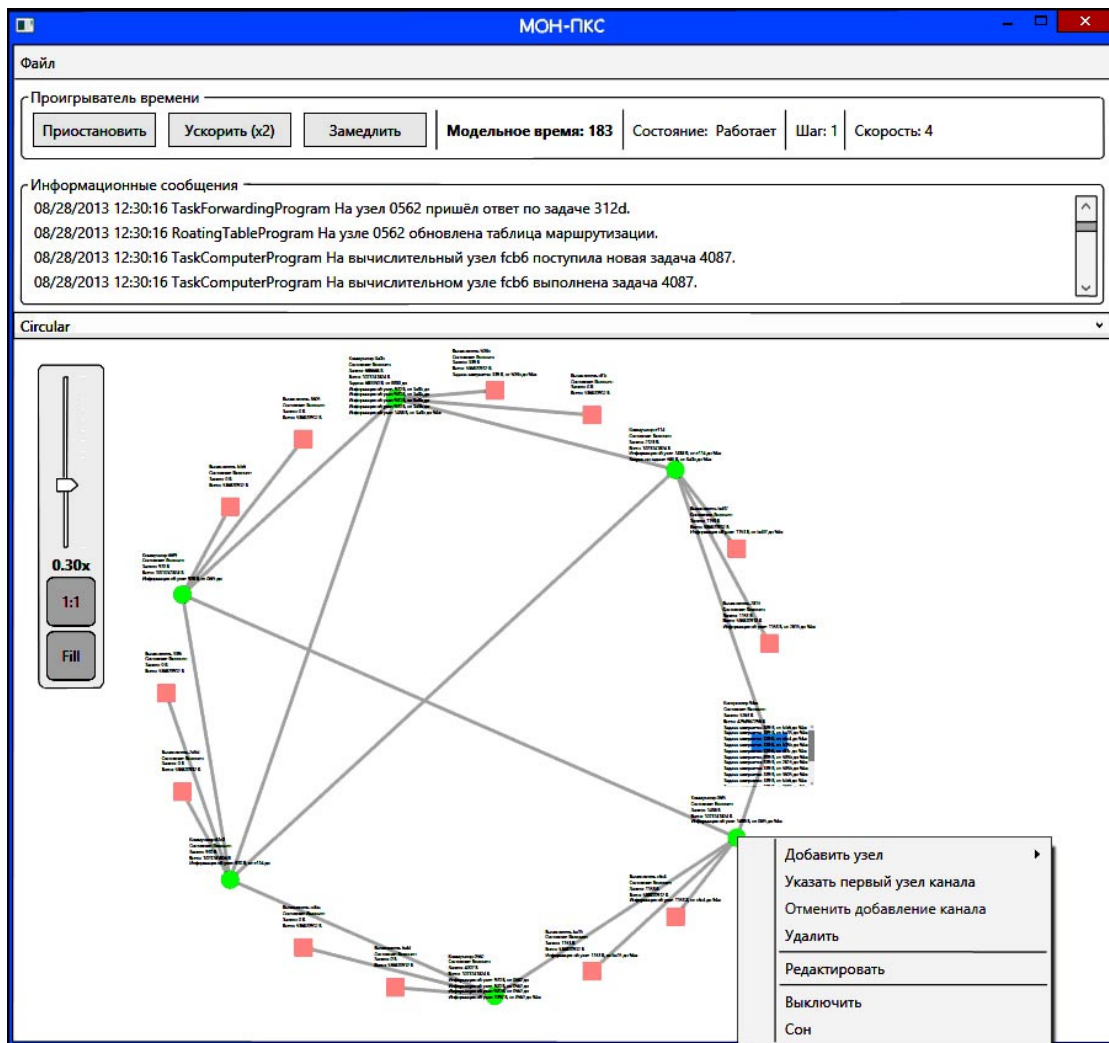


Рис. 3. Интерфейс системы

няют все текущие характеристики, в частности, списки всех выполняемых или передаваемых задач для каждого устройства сети. Таким образом, пользователь имеет возможность сохранять промежуточные результаты моделирования на различных этапах и может продолжить моделирование в следующий раз непосредственно с данного этапа, загрузив сохраненный файл.

Помимо конфигурации сети, опций сохранения и загрузки доступна возможность формирования сценария моделирования, определяющего параметры потоков поступающих в сеть задач (закон распределения задач по времени по каждому устройству), а также отдельные параметры поступающих задач (объем, вычислительная сложность, тип, приоритет и другие). После формирования сценариев моделирования возможно их сохранение и дальнейшая загрузка. Таким образом, конфигурации сети отделены от сценариев моделирования, что позволяет провести ряд разнообразных экспериментов моделирования для одной или различных топологий сети. В настоящий момент с помощью разработанного прототипа системы

продолжаются исследования преимуществ и ограничений мультиагентного подхода.

8. ЗАКЛЮЧЕНИЕ

Разрабатываемая мультиагентная система планирования задач в программно-конфигурируемых сетях основывается на динамическом улучшении состояния сети путем локальной оптимизации загрузки серверов и коммутаторов на основе «рыночного» подхода, посредством выражения всех характеристик для решения уступок через виртуальный денежный эквивалент. Агенты задач, ресурсов и каналов стремятся согласованно повысить свою виртуальную прибыль, что должно привести к улучшению характеристик системы.

В дальнейшем планируется проведение исследований динамики очередей задач в ПКС, характеристик загрузки серверов, пропускной способности каналов и влияния изменения топологии на данные параметры.

Работа выполняется в рамках государственного контракта № 14.514.11.4086.

Литература

1. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J. Openflow: Enabling innovation in campus networks // SIGCOMM CCR, 2008. Vol. 38, № 2. P. 69–74.
2. OpenFlow <http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf> (дата обращения 19.08.2013).
3. Gude N., Koponen T., Pettit J., Pfaff B., Casado M., McKeown N., Shenker S. NOX: Towards an operating system for networks // SIGCOMM CCR, 2008. Vol. 38, № 3.
4. Martina Maggio, Enrico Bini, Georgios C. Chasparis, Karl-Erik Erzén. A Game-Theoretic Resource Manager for RT Applications // Proceedings of the 25-th Euromicro Conference on Real-Time Systems, Paris, France, July 2013.
5. Zhoujun Hu, Zhigang Hu, and Zhenhua Liu. Resource Availability Evaluation in Service Grid Environment // Proceedings of the Asia-Pacific Services Computing Conference (APSCC). IEEE, 2007. P. 232–238.
6. Волкович Я.В., Граничин О.Н. Адаптивная оптимизация сервера, обрабатывающего очередь заданий // Стохастическая оптимизация в информатике, 2005. Т. 1. С. 17–28.
7. Амелина Н.О., Фрадков А.Л. Приближенный консенсус в стохастической динамической сети с неполной информацией и задержками в измерениях // Автоматика и телемеханика, 2012. № 11. С. 6–29.
8. Виттих В.А., Скобелев П.О. Метод сопряженных взаимодействий для управления распределением ресурсов в реальном масштабе времени // Автоматика, 2009. Т. 45, № 2. С. 78–87.
9. Скобелев П.О. Мультиагентные технологии в промышленных применениях: к 20-летию основания Самарской научной школы мультиагентных систем // Мехатроника, автоматизация, управление, 2010. № 12. С. 33–46.

MULTI-AGENT TASKS SCHEDULING SYSTEM FOR SOFTWARE-CONFIGURABLE NETWORKS

Abstract

The article describes the multi-agent system for tasks scheduling in software-configurable networks. The system is designed to simulate the distribution of tasks to computing resources with the consideration of network dynamic characteristics and topology.

Keywords: multi-agent system, real-time scheduling, adaptability, software-configurable network.

*Скобелев Петр Олегович,
доктор технических наук,
ведущий научный сотрудник
ИПУСС РАН, г. Самара,
Генеральный директор ООО «НПК
«Мультиагентные технологии»
petr.skobelev@gmail.com,*

*Граничин Олег Николаевич,
доктор физико-математических
наук, профессор кафедры системного
программирования математико-
механического факультета СПбГУ,
Oleg_granichin@mail.ru,*

*Будаев Денис Сергеевич,
ведущий программист ООО «НПК
«Сетецентрические Платформы»,
budaev@smartsolutions-123.ru,*

*Ларюхин Владимир Борисович,
Директор центра разработки
ООО «НПК «Разумные Решения»,
vl@smartsolutions-123.ru,*

*Майоров Игорь Владимирович,
ведущий специалист научно-
исследовательского отдела
ООО «НПК «Мультиагентные
Технологии»,
imayorov@smartsolutions-123.ru.*



Наши авторы, 2013.
Our authors, 2013.