

Разработка системы имитационного моделирования работы вычислительной среды на основе подхода ПКС

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

Ключевые слова: распределенная вычислительная среда, программно-конфигурируемые сети, мультиагентная система, планирование в реальном времени, адаптивность, имитационное моделирование.

Аннотация. Рассматривается мультиагентная система имитационного моделирования работы программно-конфигурируемой сети. Система предназначена для моделирования распределения и выполнения задач на вычислительных ресурсах с учетом динамических характеристик и топологии 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 и его структура изменились, 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.

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 им рассматривать сеть как логическую или виртуальную сущность. Разработанный протокол OpenFlow (OF), применяющийся во многих сетевых устройствах и 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 в 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 рассмотрена 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. Описание модели

Предположим, имеется набор вычислительных ресурсов (серверов), соединенных через коммутаторы в сеть, управляемую контроллером ПКС. Серверы предназначены для выполнения задач разного типа с различной скоростью, зависящей от типа задачи и от производительности вычислительного ресурса. В систему поступает поток задач, каждая из которых может быть выполнена на ограниченном подмножестве серверов (Рис. 1)

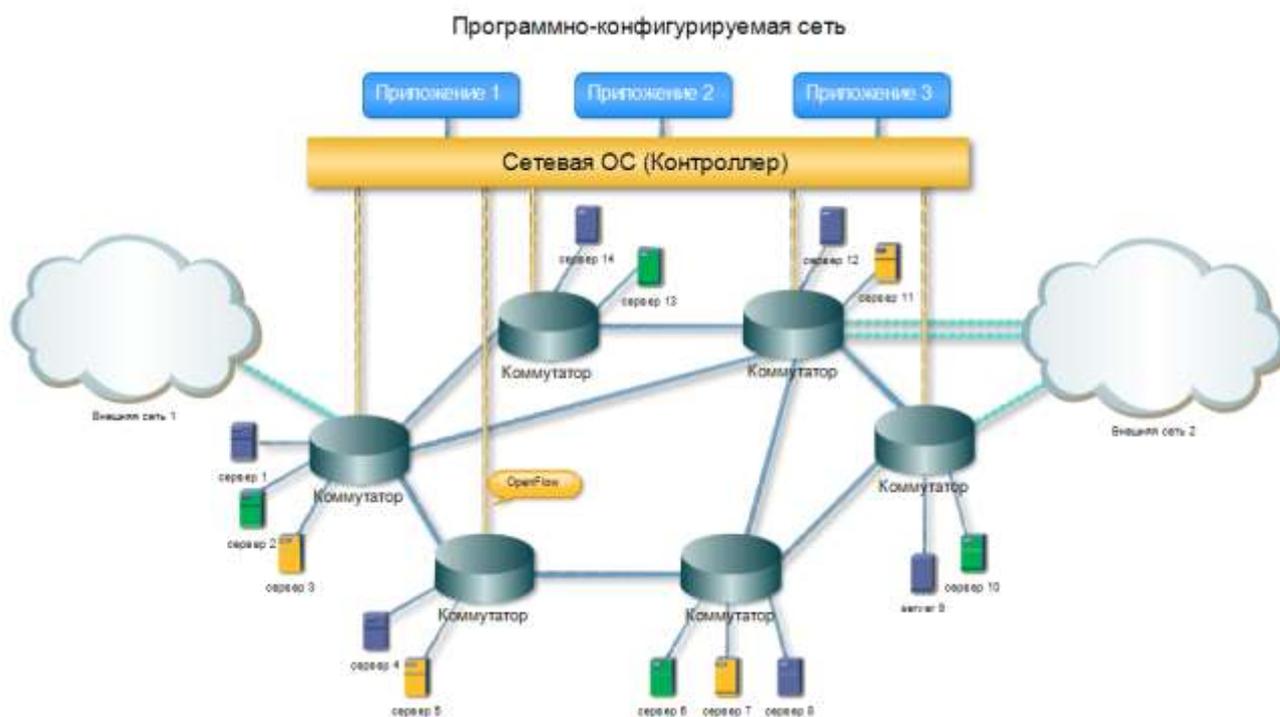


Рис. 1 – Топология сети

Соответствие серверов и типов задач определяется в описании каждой задачи. Задачи поступают в определенные моменты времени, определяемые используемым в генераторе задач потоком распределения, и должны эффективно перераспределяться по сети. 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. При необходимости, например, в случае непредвиденных отключений узлов сети, производится 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. Логика управления и моделирования разделяется между аппаратными (узлы, каналы) и программными составляющими (программные модули, агенты). Каждый агент привязан к определенному узлу и, используя предоставляемые узлом ресурсы (память, вычисления, каналы передачи данных), выполняет собственную логическую задачу, стремится к выполнению своей цели.

4. Особенности и ограничения

Примем, что с каждым коммутатором сети связаны несколько вычислительных узлов (серверов, ресурсов), которые призваны обслуживать запросы на выполнение задач. Каждая задача может поступить в общем случае на произвольные коммутаторы ПКС из внешних по отношению к рассматриваемой сетей. В один момент модельного времени по каналу передачи данных может передаваться несколько задач. Аналогично, в единицу модельного времени на вычислительном узле может выполняться несколько задач с некоторой общей суммарной трудоемкостью, которые будут разделять общую суммарную производительность данного ресурса. Таким образом, время обработки вычислителем каждой задачи определяется общим списком всех выполняемых вычислителем задач.

Состояние программно-конфигурируемой сети в целом определяется ее текущей топологией, количеством и параметрами обслуживаемых задач в очередях коммутаторов и вычислителей, передаваемыми в каналах задачами, параметрами коммутаторов и вычислителей, а также логикой работы агентов устройств сети.

Рассмотрим основные абстракции модели и их параметры:

- Узел: память изменяемого размера (хранилище сообщений), таблица маршрутизации, таблица базовой информации об узлах сети, список каналов связи. Типы узлов:
 - коммутатор;
 - вычислитель: производительность (флоп/с), типы работ, список выполняемых задач;
 - контроллер: таблица детальной информации об узлах сети, список обработанных задач.
- Канал: пропускная способность (Б/с), список передаваемых сообщений.
- Сообщение: отправитель, получатель, размер (Б).
- Задача: вычислительный размер (флоп), размер данных (Б), тип работ.

Рассматриваемая модель имеет некоторые допущения модели и ограничения:

- Задача может поступить на любой коммутатор сети в любой момент модельного времени.
- Задача атомарная и далее неделимая порция информации в модели ПКС.
- Все данные в модели доставляется с помощью сообщений, в том числе и задачи.
- Коммутатор может обслуживать несколько задач в один момент модельного времени.
- Канал может передавать несколько сообщений с один момент модельного времени.
- Ресурс может обрабатывать несколько задач в один момент модельного времени.
- Каждый узел сети представлен несколькими программными сущностями:
 - Один программный объект для моделирования аппаратной части устройства.
 - Несколько программных объектов для описания логики работы устройства.
- Программные объекты описания логики устройств представляют агентов устройств.
- Агент контроллера собирает и обрабатывает информацию от агентов устройств.
- Агенты стараются получить выгоду через переговоры и компромиссы между собой.

5. Реализация системы

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 используются библиотеки Graph# и QuickGraph.

Текущая реализация системы позволяет представить исследуемую вычислительную среду визуально в виде графа, производить начальную конфигурацию сети, а также определять параметры отдельных устройств: объемы памяти отдельных коммутаторов и серверов, пропускные способности каналов связи, производительности серверов обработки, включены или выключены устройства и т.д. (Рис. 2).

В ходе процесса моделирования возможно изменение масштаба модельного времени относительно реального, что позволяет ускорять или замедлять процесс моделирование 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 непосредственно самого процесса моделирования в любой момент времени. При этом моделируемые устройства сохраняет 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.

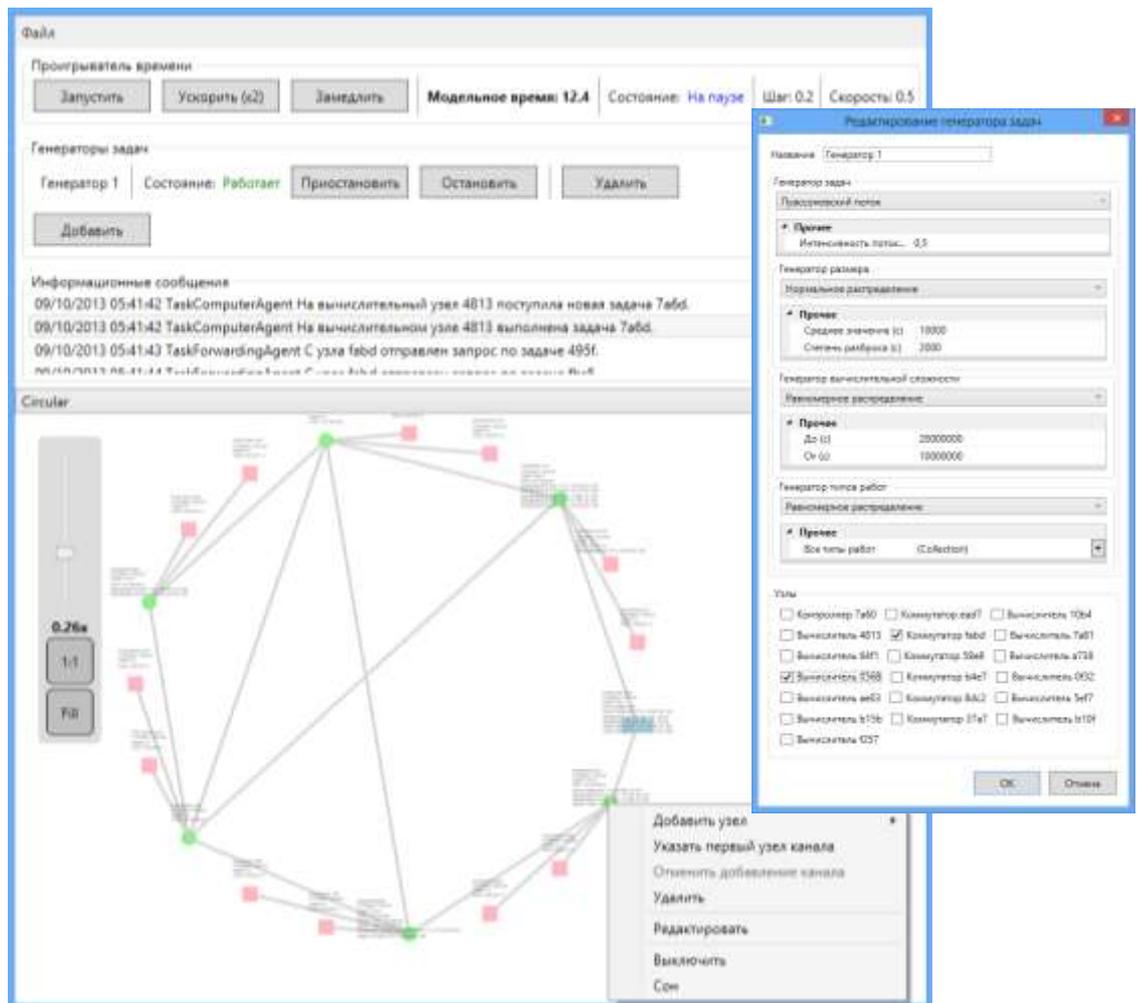


Рис. 2 – Интерфейс системы

В рамках конфигурации моделирования 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). Имеется возможность выбора из трех вариантов распределения задач по времени: равномерного, нормального, Пуассона. Для каждого типа распределения система позволяет указать параметры распределения для основных параметров задач, а также указать коммутаторы, на которые будут поступать задачи с периодичностью, определяемой потоком поступления по времени.

В настоящий момент с помощью разработанного прототипа системы выполняются исследования преимуществ и ограничений мультиагентного подхода. Для этого в прототипе реализована подсистема аналитики. В рамках данной подсистемы каждая функция анализа реализована в отдельном настраиваемом программном модуле – анализаторе. Предусмотрены возможности динамического отображения результатов работы анализатора (во время моделирования, в виде графиков), а также выгрузка результатов в файл Microsoft Excel.

6. Заключение

Разрабатываемая система планирования задач в программно-конфигурируемых сетях основывается на динамическом улучшении состояния сети путем оптимизации загрузки серверов и коммутаторов. Агенты стремятся согласованно повысить эффективность использования вычислительных ресурсов, что должно привести к улучшению характеристик системы.

В настоящий момент выполняются исследования свойств вычислительных сред, построенных на базе основных топологий сетей (звезда, кольцо и др.), анализ и совершенствование разработанных алгоритмов и моделей, улучшение производительности системы в процессе моделирования, разработка инструментов и анализаторов.

7. Литература

1. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks // SIGCOMM CCR, vol. 38, no. 2, pp. 69–74, 2008.
2. OpenFlow Switch Specification, <http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf>, последнее обращение 19.08.2013
3. N. Gude, T. Koronen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: Towards an operating system for networks // SIGCOMM CCR, vol. 38, no. 3, 2008.
4. Martina Maggio, Enrico Bini, Georgios C. Chasparis, Karl-Erik Årzén, A Game-Theoretic Resource Manager for RT Applications // Proceedings of the 25th 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. Я.В. Волкович, О.Н. Граничин, Адаптивная оптимизация сервера, обрабатывающего очередь заданий // Стохастическая оптимизация в информатике, Том 1, стр. 17-28, 2005.
7. Н.О. Амелина, А.Л. Фрадков, Приближенный консенсус в стохастической динамической сети с неполной информацией и задержками в измерениях // Автоматика и телемеханика. 2012. № 11. С. 6-29.
8. В.А. Виттих, П.О. Скобелев, Метод сопряженных взаимодействий для управления распределением ресурсов в реальном масштабе времени // Автометрия, 2009, т.45, №2 с.78
9. П.О. Скобелев. Мультиагентные технологии в промышленных применениях: к 20-летию основания Самарской научной школы мультиагентных систем // Мехатроника, автоматизация, управление. – 2010, №12.