ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ТУМАННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТАРИЯ IFOGSIM

Ворожцов Анатолий Сергеевич,
МТУСИ, доцент, к.т.н., Москва, Россия,
as.vorojcov@mail.ru

Тутова Наталья Владимировна,
МТУСИ, доцент, к.т.н., Москва, Россия,
n.v.tutova@mtuci.ru

Аннотация
iFogSim – открытый пакет программ, предназначенный для проведения имитационного моделирования туманных вычислительных сред на платформе Java. В работе приведен краткий обзор пакета: назначение, структура, а также возможности и ограничения его применения для проведения занятий по соответствующим профильным дисциплинам.

Ключевые слова
Имитационное моделирование, центры обработки данных, облачные вычисления, туманные вычисления, CloudSim, iFogSim, распределение ресурсов.

Введение

Одной из базовых технологий построения цифровой экономики является технология Интернета вещей (IoT), которая лежит в основе таких областей, как умная медицина, умный город, умный дом, умное производство и сельское хозяйство. Облачные вычисления являются фундаментом для предоставления инфраструктурных, платформенных и программных сервисов для разработки систем Интернета вещей [1,2]. Однако облачные центры обработки данных (ЦОД) могут находиться на значительном расстоянии от источников данных Интернета вещей, что увеличивает задержку передачи данных, анализ и принятие решений. Это неприемлемо для сервисов реального времени, такие как мониторинг состояния здоровья критических пациентов, аварийное пожаротушение и управление трафиком. Устройства IoT географически распределены и могут генерировать большое количество данных в единицу времени, что может привести к перегрузке каналов связи и центров обработки данных.

Туманные вычисления расширяют облачные сервисы до границ сетей, что приводит к снижению сетевых задержек за счет географического распределения компонент приложений IoT и повышают масштабируемость таких систем. Особенность обработки IoT-данных в том, что узлы (капли тумана), такие как сетевое оборудование, микро-ЦОД, нано-сервер, смартфон, персональный компьютер, располагаются близко к источнику данных. Кроме этого, по сравнению с облачными ЦОД, узлы тумана не обладают большими вычислительными ресурсами, поэтому часто облачные и туманные вычисления работают интегрированно: функция долгосрочной обработки данных, собранных с устройств IoT, передается в облако.

Управление ресурсами в туманных вычислениях относится к классу сложных процессов, поскольку оно включает значительное количество разнообразных узлов тумана и ограничений на используемые ресурсы. Дополнительные трудности вызывает интеграция распределенных приложений с облаком в совместном управлении ресурсами [1]. В процессе разработки таких систем для оценки производительности и масштабируемости, для выявления узких мест и проверки алгоритмов использования ресурсов реализация натурных экспериментов слишком дорогостоящая и часто невозможная. Эта проблема может быть решена только с использованием имитационного моделирования.

Поэтому ознакомление студентов профильных направлений со средствами имитационного моделирования туманных вычислений необходимо. Данный инструментарий может также использоваться для проведения исследований при написании магистерских и кандидатских диссертаций, а также в практической деятельности при проектировании туманных вычислительных сред.

Обзор средств имитационного моделирования туманных вычислений

В настоящее время разработано несколько систем имитационного моделирования для Интернета вещей и туманных вычислительных сред, такие как WSNet [3], SimpleIoTSimulator [4] и iFogSim [5].

WSNet – это событийный симулятор для беспроводных сетей, который также может использоваться для Интернета вещей. Он способен моделировать узлы с различными источниками энергии, моделями мобильности, радиоинтерфейсами, приложениями и протоколами маршрутизации. Моделирование окружающей среды также поддерживается WSNet; фактически, среда предлагает возможность моделирования физических явлений (например, пожара) и физических мер (например, температуры и влажности). Эти значения (например, температура может наблюдаться узлами и может также влиять на узлы.

SimpleIoTSimulator – это коммерческий симулятор для создания IoT-сред, состоящих из множества датчиков и шлюзов. В SimpleIoTSimulator поддерживает стандартные протоколы интернета вещей, в том числе CoAP и MQTT. Цель SimpleIoTSimulator – дать возможность поставщикам IoT-платформ и шлюзов улучшить качество продукции с акцентом на коммуникационные протоколы. Недостатком является то, что SimpleIoTSimulator не моделирует туманные среды, в которых службы могут быть развернуты как на пограничных, так и на облачных серверах.

Стандарт OASIS Devices Profile for Web Services

(DPWS) предназначен для обеспечения возможности развертывания веб-служб на ограниченных по ресурсам устройствах. Для ускорения разработки приложений с поддержкой DPWS в работе [6] предложен симулятор DPWSim, позволяющий разработчикам проектировать, разрабатывать и тестировать приложения Интернета вещей на основе служб с использованием технологии DPWS без наличия физических датчиков и шлюзов.

Библиотека с открытым кодом iFogSim для Java предназначена моделирования туманных вычислений. Библиотека разработана в лаборатории облачных вычислений и распределенных систем университета Мельбурна на основе платформы имитационного моделирования CloudSim, которая является одним из широко распространенных симуляторов для моделирования среды облачных вычислений [7-9]. Расширяя абстракцию базовых классов CloudSim, iFogSim позволяет моделировать вычислительную среду тумана с большим количеством узлов тумана и устройств IoT (например, датчиков, исполнительных механизмов)

Симулятор iFogSim Simulator и его компоненты

Архитектура вычислительной среды в iFogSim, представленная на рис. 1, состоит из нескольких слоев, каждый из которых отвечает за конкретные задачи, облегчающие работу более высоких слоев [5].

В архитектуре самый нижний слой состоит из устройств IoT, которые взаимодействуют с реальным миром и являются источником или приемником данных. ***Датчики IoT*** выступают в качестве источника данных для приложений и географически распределены в различных точках, воспринимая окружающую среду и передавая наблюдаемые значения на верхние слои через шлюзы для дальнейшей обработки. ***Исполнительные механизмы IoT*** работают на самом нижнем уровне архитектуры и отвечают за управление механизмом или системой. Исполнительные механизмы обычно предназначены для реагирования на изменения в окружающей среде, которые диктуются приложениями на основе информации, захваченной датчиками. Каждое устройство в IoT является либо источником, либо приемником данных и, следовательно, может быть смоделировано датчиком или исполнительные механизмом соответственно.

Рис. 1. Архитектура туманных вычислений

В работе [10] выделяют 5 типов устройств ввода данных, а именно датчики, интеллектуальные считыватели, камеры, микрофоны и коллекторы (собирающие информацию с web-сервисов или удаленных файлов). Любое устройство, принадлежащее к этому виду, имеет определенные характеристики передачи данных, например, время передачи или размер выпускаемого блока данных. В iFogSim датчик связан с определенными характеристиками входных данных, которые могут быть настроены для имитации любого типа устройства IoT, передающего данные, начиная от интеллектуальных камер и заканчивая носимыми датчиками окружающей среды и мобильными устройствами.

То же самое с исполнительными механизмами, они могут быть настроены для имитации эффектов полученной информации от приложений.

Ко второму уровню архитектуры относятся ***устройства тумана*** – это любой элемент в сети, способный размещать модули приложения. ***Шлюзы*** – это ***у***стройства тумана, которые подключают датчики к сети. Устройства тумана также включают облачные ресурсы, которые предоставляются по требованию из географически распределенных центров обработки данных. Устройства расположены в иерархической топологии «сверху-вниз» с прямой связью, возможной только между родительско-дочерней парой в иерархии, что приведено на рис. 2. Модуль приложения, работающий на устройстве тумана, отвечает за обработку всех данных, полученных от элементов, расположенных ниже устройства в иерархии. Связь «устройство с устройством» на одном уровне не поддерживается.

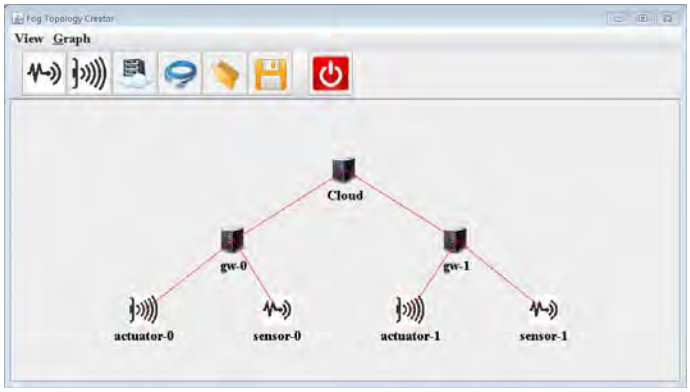

Рис. 2. Графический интерфейс для описания сетевой топологии

Следующий уровень архитектуры образуют ***потоки данных IoT***, которые представляют собой последовательности значений (кортежи-tuple), передаваемые устройствами. Эти потоки могут генерироваться датчиками, в этом случае они содержат необработанные данные, или могут быть переданы от модуля приложения к другим модулям или исполнительным механизмам. Потоки данных также генерируются устройствами тумана в виде сведений об использовании ресурсов, которые обрабатываются уровнем мониторинга для получения информации о состоянии устройств.

Уровень мониторинга отслеживает использование ресурсов, энергопотребление и доступность датчиков, исполнительных механизмов и устройств тумана. Компоненты мониторинга предоставляют эту информацию на уровень управления ресурсами и могут

предоставлять ее другим службам по мере необходимости. Более сложные компоненты уровня мониторинга, такие как прогнозирование производительности и база знаний, не были включены в текущую версию iFogSim, однако эти компоненты могут быть реализованы разработчиком модели в специально подготовленных для этого классах.

Управление ресурсами в iFogSim имеет два уровня: размещение приложений и планирование. *Политика размещения приложений* определяет, как модули приложения размещаются на устройствах тумана. Критериями размещения могут являться минимизация сквозной задержки, использование сети, затраты или потребление энергии. *Планирование ресурсов* устройств тумана в прикладных модулях формирует второй уровень управления ресурсами. Планировщик ресурсов по умолчанию равномерно распределяет ресурсы устройства между всеми активными модулями приложения. Политика планирования может быть переопределена разработчиком имитационной модели.

Прикладные (программные) модели. Приложения, разработанные для развертывания в тумане, основаны на модели распределенного потока данных (distributed data flow-DDF). Приложение моделируется как совокупность модулей, составляющих элементы обработки данных. Выходные данные, генерируемые модулем *i*, могут быть использованы в качестве входных данных другим модулем *j*, что приводит к зависимости данных между модулем *i* и *j*. эта модель приложения позволяет представить приложение в виде ориентированного графа, вершины которого представляют модули приложения, а направленные ребра обозначают поток данных между модулями.

Архитектура iFogSim поддерживает две модели, используемые для приложений Интернета вещей.

1. Модель "сенсор-процесс-исполнение" (sense-process-actuate). Информация, собранная датчиками, передается в виде потоков данных, на которые воздействуют приложения, работающие на устройствах тумана, и результирующие команды передаются на исполнительные механизмы.

2. Модель потоковой обработки. Модель потоковой обработки имеет сеть модулей приложения, работающих на устройствах тумана, которые непрерывно обрабатывают потоки данных, генерируемые датчиками. Информация для анализа и долгосрочного планирования хранится в центрах обработки данных.

Элементы архитектуры моделируются как классы iFogSim.

FogDevice. Этот класс определяет аппаратные характеристики устройств тумана и их соединения с другими устройствами, датчиками и исполнительными механизмами. Основными атрибутами класса FogDevice являются доступная память, процессор, размер хранилища, пропускная способность восходящей и нисходящей линий связи (определяющие пропускную способность устройств тумана). Методы в этом классе определяют, как ресурсы устройства тумана планируются между модулями приложения, запущенными на нем, и как модули развертываются или наоборот сворачиваются. Переопределение этих методов позволяет разработчикам подключать пользовательские политики для вышеупомянутых функций.

Sensor. Класс содержит атрибуты, представляющие характеристики датчика. Также класс содержит ссылочный атрибут на шлюзовое туманное устройство, к которому подключен датчик, и задержку соединения между ними. Кроме этого, в классе определяются выходные характеристики датчика и интервал передачи показаний, который определяет скорость прибытия кортежа на шлюзе. Установив соответствующие значения этих атрибутов, можно имитировать различные устройства, например, смарт-камеры или подключенные автомобили.

Actuator. Этот класс моделирует ***исполнительный механизм***, который приводится в действие по прибытию кортежа из модуля приложения. Он имеет свойства сетевого соединения. Атрибут в классе ссылается на шлюз, к которому подключен привод, и задержку этой связи.

Tuple. Кортежи являются единицей связи между сущностями в тумане и реализуют уровень потока данных в архитектуре. Кортежи в iFogSim представлены как экземпляры класса tuple. Кортеж характеризуется своим типом, исходными и целевыми модулями приложения. Атрибуты класса определяют требования к обработке, задаётся число миллионов инструкций и длину данных, инкапсулированных в кортеж.

Application. Структура приложения в iFogSim соответствует модели распределённого потока данных DDF, в которой приложение направленный ациклический граф, вершины которого представляют модули, обрабатывающие входные данные, и ребра, обозначающие зависимости данных между модулями. Эти сущности реализуются с помощью следующих классов.

AppModule. Экземпляры класса AppModule представляют вычислительные элементы приложений тумана и реализуют вершины графа в модели DDF. Модуль реализуется посредством расширения PowerVm класса в CloudSim. Для каждого входящего кортежа, экземпляр класса AppModule обрабатывает его и генерирует выходные кортежи, которые передаются следующим модулям в этой группе.

AppEdge. Экземпляр класса AppEdge обозначает зависимость данных между парой прикладных модулей и представляет направленное ребро в модели приложения DDF. Каждое ребро характеризуется типом кортежа, который оно передает (атрибут tupleType), а также требованиями к обработке данных, инкапсулированных в этих кортежах. AppEdge бывают двух видов – периодические (кортежи испускаются через регулярные промежутки времени) и основанные на событиях (при выполнении определенного условия).

AppLoop – это дополнительный класс, используемый для задания контуров управления процессами, представляющих интерес для пользователя. В iFogSim разработчик может указать контуры управления для измерения сквозной задержки. Экземпляр AppLoop-это, по сути, список модулей, начиная с начала контура и заканчивая модулем, где контур завершается.

Установка iFogSim

Для работы с пакетом программ iFogSim требуется виртуальная машина Java версии 8 или выше, а также интегрированная среда разработки Eclipse или NetBeans. Исходный код пакета доступен по адресу https://github.com/Cloudslab/iFogSim. Необходимо скачать архив на компьютер и включить разархивированную папку в свой Java-проект. К сожалению, для этого могут потребоваться базовые навыки работы со средой разработки. После этого можно выполнять встроенные примеры. Примеры включают в себя:

- моделирование онлайн-игры, которая работает как приложение на смартфонах и в облаке (VRGameFog.java);
- распределенная система интеллектуального видеонаблюдения (DCNSFog.java).

В обоих примерах оценивались сетевая задержка, загрузка сети и энергопотребление. Подробное описание примеров приведено в [5].

Построение имитационной модели в iFogSim

Для построения модели необходимо выполнить следующие действия.

1. Создать физические компоненты сети с определенной конфигурацией. Параметры конфигурации включают в себя память, производительность процессора в миллионах инструкций в секунду (MIPS), стоимость обработки в миллионах инструкций, пропускную способность восходящей и нисходящей линий связи, энергопотребление занятого и простаивающего устройства, а также их иерархический уровень (1—самый верхний). При создании устройств тумана более низкого уровня необходимо создать ассоциированные устройства IoT (датчики и исполнительные механизмы). При создании датчика задается интервала замеров (значение transmitDistribution). Кроме того, для создания датчиков и исполнительных механизмов требуется ссылки на идентификаторы приложения и брокера.
2. Создать логические компоненты, такие как AppModule, AppEdge и AppLoop. При создании объектов AppModule задаётся их конфигурация, объекты AppEdge включают информацию о типе кортежей, их направлении, использовании процессора и длины, а также ссылки на модуль-источник и модуль назначения. Различные типы кортежей создаются на основе данных объектов AppEdge.
3. Создать компоненты управления для определения различных политик планирования и размещения AppModule. Пользователи могут учитывать общее энергопотребление, задержку обслуживания, использование сети, эксплуатационные расходы и гетерогенность устройств при назначении приложений AppModules устройствам тумана, расширяя абстракцию класса ModuleMapping соответственно. Основываясь на информации AppEdges, требования AppModule должны быть согласованы со спецификацией соответствующего типа кортежа и удовлетворяться доступными ресурсами тумана. После проверки того, что ресурсы устройств тумана соответсвуют требованиям AppModules, информация о физических и логических компонентах пересылается на объект контроллера (Controller), который «отправляет» всю систему в ядро cloudsim для моделирования.

Для облегчения описания топологии физической сети, в iFogSim разработан графический интерфейс, который позволяет пользователю задавать физические элементы сети (устройства тумана, датчики, исполнительные механизмы и линии связи) и определять их характеристики (рис. 2). Графическую схему топологий можно сохранить и загрузить в формате JSON.

Заключение

iFogSim—это мощное средство имитационного моделирования туманных вычислительных сред Интернета вещей, функционал которого непрерывно развивается. Открытость кода данного пакета позволяет считать его перспективным для использования в учебном процессе для изучения технологии туманных вычислений.

Литература

1. *Rajkumar Buyya and Satish N. Srirama* (eds.), Fog and Edge Computing: Principles and Paradigms, 480 pages, ISBN: 978-111-95-2498-4, Wiley Press, New York, USA, January 2019.
2. *Росляков А.В., Ваняшин С.В., Гребешков А.Ю., Самсонов М.Ю.* Интернет вещей; под ред. А.В. Рослякова. Самара: ПГУТИ, ООО «Издательство Ас Гард», 2014. 340 с.
3. *Chelius G, Fraboulet A, Fleury E.* WSNET: a modular event-driven wireless network simulator, 2006.
4. SimpleIoTSimulator, https://www.smplsft.com/SimpleIoTSimulator.html, Дата посещения: 15.12.2019.
5. *Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R.* iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Softw Pract Exper. 2017;47:1275–1296.
6. *Han SN, Lee GM, Crespi N,* et al. DPWSIM: a simulation toolkit for iot applications using devices profile for web services. In 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, South Korea, 2014;544-547.
7. *Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya*. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011.
8. *Тутов А.В., Тутова Н.В., Ворожцов А.С.* Моделирование процессов распределения ресурсов в облачных центрах обработки данных // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 4. С. 76-80.
9. *Ворожцов А.С., Тутова Н.В.* Методика использования пакета программ Cloudsim в учебном процессе // Методические вопросы преподавания инфокоммуникаций в высшей школе. 2017. Т. 6. № 1. С. 13-15.
10. *Yousfi A, Bauer C, Saidi R, Dey AK.* UBPMN: a BPMN extension for modeling ubiquitous business processes. Inf Softw Technol. 2016;74:55-68.