

ОБЗОР ТЕХНОЛОГИЙ ОРГАНИЗАЦИИ ТУМАННЫХ ВЫЧИСЛЕНИЙ

© 2020 А.А. Кирсанова¹, Г.И. Радченко¹, А.Н. Черных^{1,2}

¹Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76),

²Научно-исследовательский центр Энсенады

(22860 Энсенада, Мексика, Carretera Ensenada - Tijuana No. 3918)

E-mail: alexander.a.kirsanov@susu.ru, gleb.radchenko@susu.ru, chernykh@cicese.mx

Поступила в редакцию: 14.06.2020

Поскольку Интернет вещей (IoT) становится частью нашей повседневной жизни, наблюдается быстрый рост числа подключенных устройств. Устоявшийся подход, основанный на технологиях облачных вычислений, не может обеспечить необходимое качество обслуживания в таких условиях, в частности, в вопросах уменьшения времени задержки при передаче данных. Технология туманных вычислений сегодня рассматриваются как многообещающее решение для обработки большого объема критически важных и чувствительных ко времени данных. В этой статье рассмотрена технология облачных вычислений, а также анализируются предпосылки к эволюционному развитию этого подхода и появлению концепции туманных вычислений. В рамках обзора ключевых особенностей туманных вычислений произведен разбор часто встречающейся путаницы с объединением понятий туманных и краевых вычислений. Приведен обзор технологий организации туманных вычислений: виртуализация, контейнеризация и оркестрация, а также систематический анализ наиболее популярных платформ, обеспечивающих поддержку туманных вычислений. В результате анализа нами предлагается два подхода к классификации платформ туманных вычислений: по принципу открытости/закрытости компонентов, а также трехуровневая классификация на основе предоставляемого функционала платформы (Deploy-, Platform- и Ecosystem as a Service).

Ключевые слова: облачные вычисления, туманные вычисления, краевые вычисления, интернет вещей.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Кирсанова А.А., Радченко Г.И., Черных А.Н. Обзор технологий организации туманных вычислений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2020. Т. 9, № 3. С. 35–63. DOI: 10.14529/cmse200303.

Введение

В нынешнюю эпоху данные являются основным товаром, а наличие большего количества данных и возможность их эффективного интеллектуального анализа создает большую ценность для предприятий, управляемых на основе данных [26]. По данным Международной корпорации данных (IDC), объем генерируемых цифровых данных в 2010 году превысил 1 зеттабайт [60]. Кроме того, с 2012 года ежедневно генерируется 2,5 эксабайта новых данных [49]. По оценкам Cisco, к 2020 году будет около 50 миллиардов подключенных устройств [16]. Эти подключенные устройства составляют *Интернет вещей* (англ. *Internet of Things — IoT*) и генерируют огромное количество данных в реальном времени. Современные архитектуры мобильных сетей уже сейчас планируются с учетом тех нагрузок, которые возникают при передаче и обработке таких астрономических объемов данных.

В текущих реализациях облачных приложений большая часть данных, требующих хранения, анализа и принятия решений, отправляется в центры обработки данных в облаке [59]. По мере увеличения объема данных, перемещение информации между IoT-устройством и облаком может быть неэффективным или даже невозможным в некоторых случаях из-за ограничений пропускной способности или требований к латентности

вычислительной сети. По мере появления приложений, чувствительных ко времени отклика (таких как мониторинг пациентов, автомобили с автоматическим управлением и другое), удаленное облако не сможет удовлетворить потребность этих приложений в обеспечении сверхнадежной связи с минимальной задержкой [81]. Более того, в некоторых приложениях отправка данных в облако может оказаться невозможной из-за проблем конфиденциальности.

Для решения проблем приложений, требующих высокой пропускной способности вычислительной сети, возможности работы с географически рассредоточенными источниками данных, сверхнизкими задержками и обеспечением локальности обработки данных существует типичная потребность в вычислительной парадигме, которая обеспечивала бы универсальный подход к организации вычислений как в облаке, так и на базе вычислительных узлов ближе к подключенным устройствам. Концепция *туманных вычислений* (англ. *Fog computing*) была предложена индустрией и научным сообществом для устранения разрыва между облаком и устройствами IoT путем предоставления вычислительных возможностей, хранения, организации сетевого взаимодействия и управления данными на узлах сети, расположенных в непосредственной близости от устройств IoT [4, 54]. Исследовательское сообщество предложило ряд подобных вычислительных парадигм для решения упомянутых проблем, таких как *краевые вычисления* (англ. *Edge computing*), *мгlistые вычисления* (англ. *Mist computing*), *росистые вычисления* (англ. *Dew computing*) и другие. В этом обзоре мы рассматриваем туманные вычисления и технологии их обеспечения и утверждаем, что туманные вычисления являются более общей формой вычислений, главным образом, из-за их всеобъемлющей области определения и гибкости. Также, мы представляем анализ наиболее популярных платформ, обеспечивающих поддержку туманных вычислений, в части базовых технологий их организации и предоставляемых сервисов. На основе проведенного анализа мы предлагаем два подхода к классификации платформ туманных вычислений: по принципу открытости/закрытости компонентов, а также трехуровневую классификацию на основе предоставляемого функционала платформы (Deploy-, Platform- и Ecosystem as a Service).

Статья организована следующим образом. В разделе 1 рассмотрены облачные вычисления как основа новых вычислительных концепций, предпосылки появления облачных вычислений, их ключевые характеристики и предпосылки к появлению новых вычислительных концепций. Раздел 2 посвящен туманным и краевым вычислениям, их истории появления, а также определению и их ключевым характеристикам. В разделе 3 рассматриваются технологии, обеспечивающие поддержку туманных вычислений: виртуализация и оркестрация. В разделе 4 представлен обзор платформ туманных вычислений: частных, публичных, с открытым исходным кодом, а также сделано предложение о классификации туманных платформ. В заключении приводится краткая сводка результатов, полученных в рамках данного исследования, и указаны направления дальнейших исследований.

1. Облачные вычисления как основа новых вычислительных концепций

1.1. Предпосылки появления облачных вычислений

Наиболее ранним предком облачных технологий принято считать зародившуюся в 1960-е годы концепцию *коммунальных вычислений* (англ. *utility computing*) [18, 75]. Подход коммунальных вычислений подразумевал предоставление вычислительных ресурсов

по тем же принципам, по которым предоставляются такие коммунальные услуги как водо-, электро- и газоснабжение. Идея была призвана обеспечить пользователей «вычислениями по требованию» для получения максимальной эффективности при гибком ценообразовании [46]. В пакет предоставляемых услуг включались различные типы ресурсов: собственно вычислительные ресурсы, ресурсы хранения данных и другие вычислительные сервисы. Данные идеи описали Джон МакКарти [75] и Дуглас Пархил [66], который описал практически все основные характеристики существующих сегодня облаков, а также впервые употребив сравнение с электрической сетью.

Данная концепция не нашла широкого распространения вплоть до 90-х годов XX века из-за технических сложностей, возникающих при развертывании и использовании данной архитектуры [3, 6, 27, 44, 46, 75]. Основной причиной аналитики называют недостатки организации сетей того времени, не обеспечивающих достаточной пропускной способности для реализации коммунальных вычислений [30].

В середине 90-х годов, совершенствование сетевых технологий и увеличение скорости передачи данных, привело к новому витку исследований коммунальных вычислений, но уже в рамках понятия *грид вычислений* (англ. *Grid computing* от англ. *grid* — сеть) [20, 27, 44] — по аналогии с *electric power grid* — электрической сетью. Идея коммунальных вычислений в рамках грид вычислений претерпела преобразование [24, 61] и под грид-вычислениями стали понимать объединение ресурсов отдельных высокопроизводительных вычислительных систем в единую сеть.

В [72] указываются следующие ключевые характеристики грид-вычислений.

- *Децентрализованная координация ресурсов.* Грид-система должна объединять и координировать ресурсы и пользователей в разных частях сети и решать вопросы политики безопасности, оплаты и членства.
- *Стандарты и протоколы* с открытым исходным кодом должны использоваться для аутентификации, авторизации, обнаружения ресурсов и доступа к ресурсам грид-системы.
- *Качество предоставления услуг:* ресурсы должны использоваться скоординированным образом для обеспечения необходимого качества обслуживания, времени отклика, пропускной способности и доступности для удовлетворения сложных требований пользователей.

В качестве недостатков такого подхода, можно выделить то, что принципиальная децентрализованность грид-систем, отсутствие единого сетевого адресного пространства и единого администратора базовой вычислительной инфраструктуры усложняет обеспечение отказоустойчивости и динамическое управление ресурсами, доступными конечным пользователям [29, 41, 61]. Данные недостатки обусловили дальнейшее эволюционное развитие и появление облачных вычислений, которые зачастую используют модель грид-вычислений для расширения вычислительных ресурсов [41].

1.2. Ключевые характеристики облачных вычислений

Развитие облачного рынка таким, каким мы его знаем сегодня, стало возможным с появлением Amazon Web Services (AWS) в 2002 году [53]. В 2006 году Amazon запустил Elastic Compute Cloud (EC2) в качестве коммерческого веб-сервиса, который позволял небольшим компаниям и отдельным лицам арендовать часть ИТ-инфраструктуры, на которой можно запускать любые приложения [14]. Именно данным шагом Amazon обеспечил дальнейшую популяризацию облачных вычислений. На сегодняшний день компания Amazon предоставляет полный пакет облачных инфраструктурных услуг, включая

хранение, вычисление и даже возможности человеческого интеллекта через Amazon Mechanical Turk.

В 2011 году Национальный институт стандартов и технологий США (The National Institute of Standards and Technology, NIST) опубликовал определение облачных вычислений, их основные характеристики, а также модели их развертывания и обслуживания. Так, *облачные вычисления* (англ. *Cloud computing*) определены как модель для обеспечения повсеместного, удобного сетевого доступа по требованию к общему пулу настраиваемых вычислительных ресурсов (например, сетей, серверов, хранилищ, приложений и услуг), которые могут быть быстро предоставлены и освобождены с минимальными усилиями по управлению или взаимодействию с поставщиком услуг [50]. В рамках данного документа отмечается, что ключевыми характеристиками облачных вычислений являются:

- **самообслуживание по требованию** — потребитель самостоятельно определяет свои вычислительные потребности: серверное время, скорости доступа и обработки данных, объем хранимых данных, без необходимости прямого взаимодействия с представителем поставщика услуг. Так, в статьях [40, 52, 79] приведены примеры решений, обеспечивающих повышение быстродействия в системах облачных вычислений с обеспечением самообслуживания по требованию;
- **универсальный доступ по сети** — услуги облачных систем доступны потребителям по сети передачи данных вне зависимости от используемого терминального устройства;
- **объединение ресурсов** — поставщик услуг объединяет ресурсы вычислительной системы в единый пул, обеспечивая возможность динамического перераспределения мощностей между большим числом независимых потребителей в условиях постоянного изменения спроса на мощности. При этом, потребители контролируют только основные параметры запрашиваемых услуг (например, объем необходимой памяти, скорость доступа), но фактическое распределение ресурсов, предоставляемых потребителю, осуществляет поставщик. Эффективные механизмы объединения ресурсов позволяют оптимизировать расход ресурсов памяти, хранения и передачи данных в облачных системах от 50% до 65% [78].
- **эластичность** — облачные услуги могут быть предоставлены, расширены, сужены в любой момент времени, без дополнительных издержек на взаимодействие с поставщиком, как правило, в автоматическом режиме [6, 41, 83];
- **учет потребления** — поставщик услуг автоматически исчисляет потребленные ресурсы на определенном уровне абстракции (например, объем хранимых данных, пропускная способность, количество пользователей, количество транзакций) и на основе этих данных оценивает объем предоставленных потребителям услуг. Такой подход, как отмечено в работе [42], позволяет получить эффективные вычислительные мощности с гибкой ценой в зависимости от предъявляемых требований.

С точки зрения потребителя эти характеристики позволяют получить услуги с *высоким уровнем доступности* (англ. *high availability*) и низкими рисками неработоспособности, обеспечить быстрое масштабирование вычислительной системы благодаря эластичности без необходимости создания, обслуживания и модернизации собственной аппаратной инфраструктуры [29, 41, 61].

Период попыток внедрения облачных вычислений для различных целей и требований, привел к выделению следующих моделей развертывания облачных систем: **частное облако**, **публичное облако**, а также **гибридное облако** [15, 76].

Частное облако разворачивается в рамках одной организации, доступно только внутренним пользователям и не предоставляет свои ресурсы пользователям вне этой организации. **Публичное облако** разворачивается сторонними организациями и предоставляет свои ресурсы внешним пользователям на условиях договора на право пользования. **Гибридное облако** является комбинацией из двух выше описанных типов развертывания, что позволяет выстраивать баланс между частными и публичными вычислениями [15].

Частные облака хороши тем, что чаще всего физически разворачиваются как можно ближе к конечному пользователю облака, что снижает время отклика вычислительного узла и повышению быстродействия передачи данных между узлами системы. Однако частное облако настраивается исключительно под вычислительные потребности своего владельца, что одновременно является как плюсом, так и минусом частных облаков. Не каждая организация имеет достаточно ресурсов для содержания собственного частного облака, которое должно удовлетворять как техническим требованиям по доступности и надежности, так и требованиям закона того государства, на территории которого находится как облако, так и сама организация [31, 62].

Пользователи публичных облаков, в свою очередь, часто сталкиваются с проблемой отсутствия прямого контроля над базовой вычислительной инфраструктурой. Это может привести к целому ряду проблем, таких как неконтролируемый доступ третьих лиц к приватным данным, размещенным в публичном облаке; блокировка серверов пользователя, которые могут быть развернуты в одной подсети с узлами, заблокированными в тех или иных государствах; неопределенность в качестве доступных ресурсов, так как они разворачиваются на серверах, используемых совместно с третьими лицами [31]. Также встает вопрос миграции и конвертации данных при необходимости смены провайдера, предоставляющего облако в пользование.

В связи с этими недостатками каждого типа развертывания, часто провайдеры, предоставляющие облака частным компаниям, разворачивают именно гибридные облака [67], которые по требованию могут вести себя, как частные или публичные, что снимает проблемы с задержками передачи данными, безопасностью и вопросами миграции, а также эластичной настройки вычислительных ресурсов под каждую требуемую задачу.

1.3. Предпосылки к появлению новых вычислительных концепций

Несмотря на все существенные достоинства, гарантируемые публичными облачными платформами, в последние 5 лет активно стали появляться задачи, которые не могут быть эффективно решены этими подходами [21]. Так, большое число пользователей мобильных приложений, «умных» систем, таких как «умный дом», «умное предприятие», «умный город» и других IoT-решений, не всегда могут быть удовлетворены качеством услуг, предоставляемых облачными решениями, в частности, из-за увеличения объема пересылаемых данных между пользователем/устройством и облаком [33].

Появление подхода «умных» домов, производств, городов и др., наполненных множеством датчиков, исполнительных механизмов и других систем привело к пересмотру концепции архитектуры систем сбора и анализа данных. Концепция *интернета вещей* требует новых подходов к решениям хранения и быстрой обработки данных, а также возможностей быстрого отклика на изменение состояния конечных устройств [55, 56, 80]. Также, распространение мобильных устройств в качестве основных платформ для клиентских приложений затрудняет передачу и обработку большого количества данных

без возникновения проблем с задержками отклика, в связи с постоянным перемещением мобильных устройств.

С увеличением объема пересылаемых данных между устройствами IoT, клиентами и облаком, появляются проблемы увеличения времени отклика, связанные с ограничениями физической ширины сетевых каналов [46]. С другой стороны, появились чувствительные к времени отклика приложения и устройства, такие как системы жизнеобеспечения, машины-автопилоты, дроны и другие. В этих условиях удаленное централизованное облако стало неспособно удовлетворить требованиям сверхнизких временных задержек [80]. Также передача данных через множество плузов и подсетей поднимает вопрос о передаче чувствительных к конфиденциальности данных [36].

В ответ на данные проблемы частные предприятия и академическое сообщество подняли вопрос о необходимости разработки вычислительной парадигмы, удовлетворяющей требованиям новых концепций, таких как IoT [4, 47, 56]. Данная парадигма должна была заполнить разрыв между облаком и конечными устройствами, обеспечив вычисления, хранение и передачу в промежуточных сетевых узлах, наиболее приближенных к конечным устройствам. На данный момент разработано и применяется несколько парадигм, решающих данную проблему, включая туманные (fog) и краевые (edge) вычисления [12]. Каждая из этих парадигм имеет свои особенности, но все они сводятся к общему принципу — уменьшение временных задержек обработки и передачи данных, за счет переноса вычислительных задач ближе к конечному устройству.

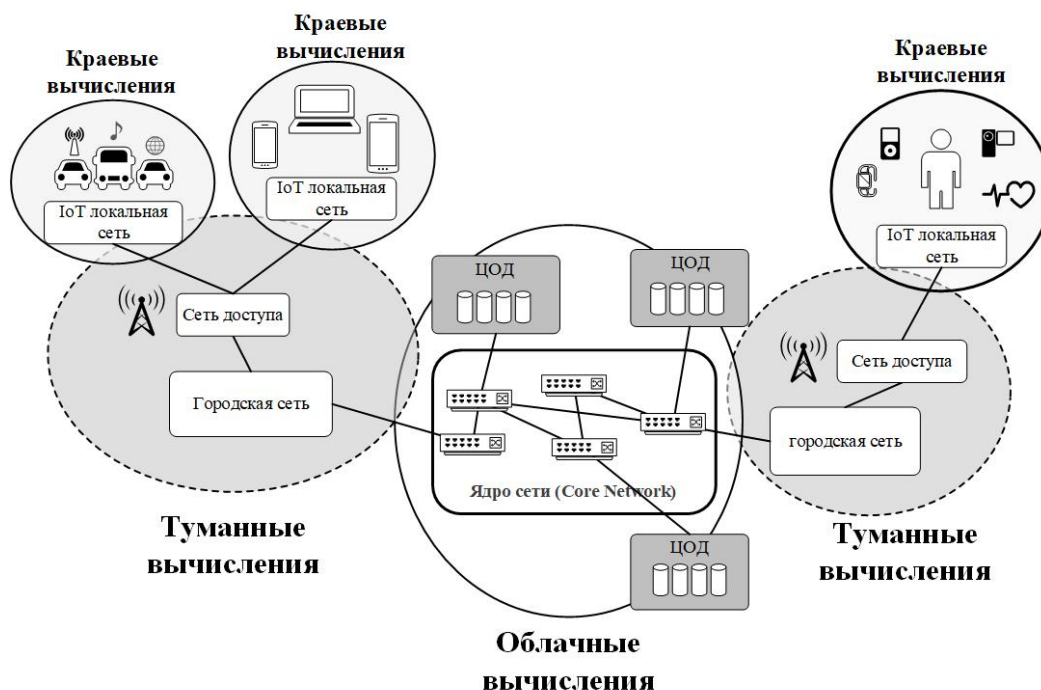


Рис. 1. Схема размещения облачных, туманных и краевых вычислений

На рис. 1 изображена диаграмма относительного распределения вычислительных ресурсов, определяемых концепциями краевых, туманных и облачных вычислений. В соответствии с ней, облачные вычисления представляют собой отдельный центр обработки данных (ЦДО) или же сеть ЦОД, расположенных далеко от пользователя, но обеспечивающих высокие вычислительные возможности. С другой стороны, краевые вычисления расположены непосредственно на краю вычислительной системы и обеспечивают небольшие вычислительные возможности, но в непосредственной близости к потребителю этих ресурсов. Туманные вычисления располагаются между краем сети и об-

лачным ЦОД, обеспечивая существенные вычислительные ресурсы близко к конечному пользователю, которые, с другой стороны, не сопоставимы с общим объемом облачных вычислений, но могут быть настраиваемы и масштабируемы в зависимости от задач конечного пользователя.

В данной статье будут рассматриваться туманные вычисления как отправная точка для появления остальных парадигм [36]. Также будут рассмотрены отличительные особенности краевых вычислений, с которыми часто объединяют туманные.

2. Туманные и краевые вычисления

2.1. История и определение

Первым из вариантов технологии для устранения проблем с временными задержками стала технология *туманных вычислений*. Термин «Туманные вычисления» впервые был предложен компанией CISCO в 2002 году [28] и был описан как «расширение для облачных вычислений, обеспечивающее предоставление вычислительных и сетевых ресурсов, а также ресурсов для хранения информации, расположенных между конечными устройствами и облачными вычислительными центрами» [4]. В 2015 году был основан консорциум OpenFog — группа компаний и академических организаций, таких как Cisco, Dell, Intel и Microsoft Corp, а также Принстонского университета, ориентированная на стандартизацию туманных вычислений (18 декабря 2018 консорциум OpenFog стал частью The Industrial Internet Consortium) [92].

В 2018 году Национальный институт стандартов и технологий США сформулировал официальное определение термина туманные вычисления:

«Туманные вычисления (ТВ) — это многоуровневая модель, обеспечивающая повсеместный доступ к общей совокупности масштабируемых вычислительных ресурсов. Модель ТВ облегчает развертывание распределенных приложений и услуг, учитывающих сетевые задержки, и состоит из *туманных узлов* (англ. *fog nodes*) (физических или виртуальных), располагающихся между умными конечными устройствами и централизованными (облачными) сервисами. *Туманные узлы* являются контекстно-зависимыми и поддерживают единую систему управления данными и организации связи. Они могут быть организованы в кластеры вертикально (для поддержки изоляции), горизонтально (для поддержки федераций сервисов) или в привязке к сетевой близости от конечных умных устройств. Туманные вычисления минимизируют время сетевого отклика поддерживаемых приложений а также обеспечивают конечные устройства локальными вычислительными ресурсами и, при необходимости, сетевым подключением к централизованным сервисам» [36].

Устранение разрыва между облаком и конечными устройствами за счет вычисления, хранения и управления данными не только в облаке, но и на промежуточных узлах [45] расширило область применения туманных вычислений, что позволило применять их в новых задачах: IoT, транспортные средства [34], «умный» город [11], здравоохранение [23], «умная» доставка (в том числе с использованием беспилотников) [74], подземная съемка в реальном времени, видеонаблюдение и др. [82]

2.2. Ключевые характеристики туманных вычислений

Из-за позднего отделения понятий туманных и краевых вычислений многие компании вводили собственные характеристики [1] и определения для туманных и краевых вычислений, зачастую объединяя их в одно [45]. В табл. 1 представлены ключевые характеристики, которые выделялись различными авторами для туманных и краевых вычислений.

В 2017 году консорциум OpenFog выпустил эталонную архитектуру туманных вычислений, которая построена на восьми основных принципах: программируемость, иерархия, гибкость, удобство обслуживания (Reliability, availability and serviceability — RAS), доступность, надежность, автономность, открытость и безопасность [54].

Таблица 1

Характеристики туманных вычислений [51]

Кто предложил характеристику	Характеристика
Bonomi et al. [68]	Высокая степень виртуализации
	Размещается между конечными устройствами и облаком
	Может быть расположено не только на самом краю сети
Cisco Systems [37]	Расширяет Облако
	Чаще используется в сфере IoT
	Может быть развернуто, где угодно
	Туманные устройства отвечают за обработку, хранение и подключение к сети
Vaquero and Rodero–Merino [86]	Гетерогенное, вездесущее и децентрализованное общение устройств между собой
	Хранение и обработка данных выполняются без использования сторонних устройств и ресурсов
	Выполнение процессов на устройстве в режиме песочницы
	«Аренда» пользовательских устройств для нужд поддержки системы
IBM [3]	Определили Туманные и Краевые вычисления как единую концепцию
	Не зависит от централизованного облака
	Располагается на конечных точках сети
	Размещение некоторых вычислительных ресурсов на краю облака
Обобщение [51]	Может использовать как виртуализацию, так и другие средства развертывания
	Обращение в облако для выполнения нечувствительных к временным задержкам операций и хранения информации
	Любое краевое устройство с достаточными вычислительными мощностями и объемом для хранения данных может выступать Туманным узлом
	Всегда размещается между конечным пользователем и облаком

В [36] выделены следующие ключевые характеристики туманных вычислений.

- **Осведомленность о местоположении и низкая латентность** — туманные узлы осведомлены о логическом расположении друг относительно друга, что позволяет производить расчет временных затрат на связь с другими узлами.
- **Географическая распределенность** — туманные сервисы и приложения способны работать с расположенными в различных географических точках шлюзами, через которые и осуществляется подключение в туман.

- **Поддержка разнородных данных** — поддержка сбора и обработки данных различных форматов, полученных с помощью различных типов сетевых коммуникационных возможностей.
- **Операционная совместимость и федеративность** — компоненты туманных вычислений должны быть способны взаимодействовать друг с другом вне зависимости от различий между собой, а сервисы должны быть распределены по различным доменам для обеспечения доступа.
- **Взаимодействия в реальном времени** — туманные приложения должны иметь возможность работать в режиме реального времени, а не с использованием пакетной обработки запросов.
- **Масштабируемость и динамичность** федеративных, туманно-узловых кластеров — туманные вычисления должны быть адаптивными по своей сути. Должна быть обеспечена поддержка следующих ключевых механизмов адаптации: эластичность вычислений, объединение возможностей ресурсов, подстройка под изменения в нагрузке данных и изменения состояния сети.

2.3. Понятия туманных и краевых вычислений

В некоторых источниках туманные вычисления называют краевыми или граничными вычислениями, основываясь на ключевом описании технологии, что сбор и анализ данных организован не в централизованном облаке, а как можно ближе к конечному устройству, «на краю сети» [4, 21, 33, 36]. Однако в [80] указано, что хотя туманные и краевые вычисления перемещают вычисления и хранение данных на край сети, ближе к конечным узлам, эти парадигмы не идентичны.

В парадигме туманных вычислений туманные узлы размещаются на границе локальной сети, зачастую они разворачиваются на базе роутеров, беспроводных точек доступа (если данные устройства поддерживают требуемые технологии для разворачивания туманного узла) [74]. В отличие от туманных вычислений, краевые вычисления размещаются еще «ближе» к конечным устройствам, уже внутри самой локальной сети на промежуточных точках доступа, а иногда и сами конечные устройства могут выступать краевыми вычислительными узлами: смартфоны, планшеты, другие вычислительные устройства с достаточными вычислительными возможностями и поддержкой разворачивания вычислительных узлов [70]. Однако это в то же самое время ограничивает их вычислительные мощности, и поэтому имеются некоторые ограничения в области их применения. На данный момент краевые вычисления применяются для решения таких задач как видеонаблюдение, кэширование видео и контроль трафика [80].

Консорциум OpenFog заявляет, что граничные вычисления часто ошибочно называют туманными вычислениями, и определяют, что основным их отличием является то, что туманные вычисления являются иерархическими и обеспечивают вычисления, создание сетей, хранение, управление и ускорение в любом месте — от облака до конечных узлов интернета вещей; в то время как краевые вычисления имеют тенденцию ограничиваться вычислениями на узлах конечных пользователей [48]. Кроме того, в [9] о туманных и краевых вычислениях авторы отмечают, что «туман включает облако, ядро, транспорт, край, клиентов и все остальное» и «туман стремится реализовать непрерывность вычислительных услуг от облака до устройств, а не рассматривать границы сети как изолированные вычислительные платформы».

Таким образом, термин «краевые вычисления» преимущественно используется в телекоммуникационной отрасли и обычно относится к базовым станциям 4G/5G, RAN (Radio Access Network) и ISP (Internet Service Provider) [9, 43]. Тем не менее, данный

термин стал с недавних пор использоваться в предметной области IoT [22, 43, 84] по отношению к локальной сети, где расположены датчики и устройства IoT. Другими словами, «краевые вычисления» расположены в пределах первого от самого IoT устройства транзитного участка сети, например, на точках доступа WiFi или шлюзах.

3. Технологии, обеспечивающие поддержку туманных и краевых вычислений

3.1. Виртуализация

Облака размещаются в ЦОД, где оборудование рассчитано на сверхбольшие нагрузки. Однако не всегда серверное оборудование загружено в полной мере, что не позволяет использовать его эффективно. Ключевой технологией, которая обеспечила поддержку облачных, а затем и туманных вычислений стала технология **виртуализации** [69], которая позволяет использовать ресурсы одной физической машины несколькими логическими виртуальными машинами (VM) на уровне слоя аппаратных абстракций (англ. Hardware Abstraction Layer — HAL). Технология виртуализации использует **гипервизор** — программный слой, обеспечивающий работу виртуальных машин на базе аппаратных ресурсов. Машина с гипервизором называется хост-машиной. Виртуальная машина, выполняемая на хост-машине, называется гостевой машиной, на которой в свою очередь могут быть установлены гостевые операционные системы (ОС). Данный вид виртуализации называют **виртуализацией на основе гипервизора**.

Также существует **виртуализация на основе контейнеров** [13], которые представляют собой упакованный, автономный, развертываемый набор прикладных компонентов, которые могут также включать промежуточное программное обеспечение и бизнес-логику в виде бинарных файлов и библиотек для запуска приложений.

В работе [58] представлен сравнительный анализ обоих типов виртуализации, на основе которого можно выделить некоторые преимущества виртуализации на основе контейнеров.

- **Аппаратные ресурсы.** Виртуализация на основе контейнеров снижает затраты на оборудование за счет возможности консолидации. Это позволяет параллельному программному обеспечению воспользоваться преимуществами истинного параллелизма, обеспечиваемого многоядерной аппаратной архитектурой.
- **Масштабируемость.** Система управления контейнерами может эффективно управлять большим количеством контейнеров, позволяя создавать дополнительные контейнеры по мере необходимости.
- **Пространственная изоляция.** Контейнеры поддерживают легкую пространственную изоляцию, предоставляя каждому контейнеру свои собственные ресурсы (например, ядро процессора, память и доступ к сети) и специфические для контейнера пространства имен.
- **Хранение.** Контейнеры имеют малый вес по сравнению с виртуальными машинами. Приложения внутри контейнеров имеют общие двоичные файлы и библиотеки.
- **Производительность.** По сравнению с виртуальными машинами, контейнеры имеют более высокую производительность (сквозную), т.к. они не эмулируют оборудование.
- **Портативность.** Контейнеры поддерживают легкую переносимость из среды разработки в производственную среду, особенно для облачных приложений.

Таким образом, существуют две основные технологии виртуализации, которые применяются в данный момент для обеспечения поддержки туманных вычислений [39].

Причем виртуализация на основе контейнеров получает все большее распространение, благодаря меньшим требованиям к производительности аппаратного обеспечения, чтобы обеспечить развертывание вычислительных узлов на промежуточных устройствах, которые могут не обладать высокими вычислительными мощностями, что особенно актуально для краевых вычислений, т.к. они запускаются даже не на самих IoT устройствах [57], а на промежуточных точках доступа, наиболее близких к устройствам.

3.2. Концепция оркестрации туманных вычислений

Когда контейнеризация стала развиваться как одна из технологий поддержки туманных вычислений, встал вопрос управления вычислительной нагрузкой для обеспечения эффективного использования географически-распределенных ресурсов [38]. Реализация туманных вычислений требует решения проблемы управления вычислительными ресурсами на другом уровне, по сравнению, например, с облачными вычислениями [73].

Первая сложная задача, которая возникает при работе с туманными вычислениями в отличие от облачных, — это управление распределением вычислительной нагрузки (**оркестрация**) между узлами тумана [43, 44] посредством размещения на них туманных сервисов, а также оркестрация этих сервисов, т.е. обеспечение эффективной совместной работы вычислительных сервисов для решения задач, возложенных на туманную среду.

В работе [77] сформулировано, что оркестрация в применении к туманным вычислениям обеспечивает централизованный механизм поиска и обнаружения туманных ресурсов, привязку приложений на основе их требований к физическим ресурсам (развертывание и планирование); управление выполнением рабочей нагрузки с контролем качества обслуживания (QoS).

Рассмотрим ключевые задачи, которые должны решаться системой оркестрации туманных вычислений (Fog Orchestrator) [5, 73].

- **Планирование заданий.** Система оркестрации должна организовывать эффективную совместную работу туманных узлов для выполнения вычислительных заданий. Брокер вычислительных ресурсов должен учитывать специфику рабочего процесса выполнения каждого задания для оптимизации процессов их выполнения.
- **Вычисление и оптимизация маршрутов.** Система оркестрации должна учитывать распределенный характер туманной среды при построении сетевых маршрутов. Она должна обеспечивать поддержку сквозного соединения узлов в случае отсутствия прямого соединения, адаптацию к динамическим сетевым топологиям, максимизацию пропускной способности сети и производительности приложений, обеспечение устойчивости сети.
- **Обнаружение.** Должно обеспечиваться обнаружение физических и виртуальных туманных устройств, а также ресурсов, связанных с ними.
- **Функциональная совместимость.** Оркестратор туманной вычислительной среды должен поддерживать единый набор протоколов, стандартных интерфейсов и онтологий. Это обеспечит возможность различным узлам и приложениям в системе взаимодействовать друг с другом.
- **Минимизация задержек.** Одна из основных задач оркестратора туманных вычислений — уменьшение задержек передачи и обработки данных. Добиться этого позволяют интеллектуальные механизмы оптимизации потоков данных и планирования ресурсов.

- **Отказоустойчивость.** Должно обеспечиваться беспшовное взаимодействие между всеми участниками туманной среды, вне зависимости от возможных сбоев и проблем как на физическом, так и логическом уровне.
- **Прогнозирование и оптимизация.** За счет сбора, хранения и анализа работы узлов системы, система оркестрации может обеспечивать оптимизацию маршрутов передачи данных и взаимодействия устройств между собой для выполнения требований по задержкам и качеству предоставляемых услуг.
- **Безопасность и приватность.** Из-за усложнения топологии туманной сети существенно меняется постановка задачи обеспечения безопасности и конфиденциальности обрабатываемых данных.
- **Аутентификация и контроль доступа.** Решение вопросов безопасности приводит к введению учетных записей, ролей и прав доступа к каждому отдельному узлу или группе узлов для управления, развертывания и использования.

4. Обзор платформ туманных вычислений

При обзоре существующих платформ для развертывания туманных вычислений были рассмотрены коммерческие платформы, а также платформы с открытым исходным кодом. Сложность анализа коммерческих платформ заключается в отсутствии информации об их архитектуре и применяемых технических решениях, которые составляют коммерческую тайну. Однако анализ коммерческих решений показал, что среди коммерческих туманных платформ встречаются как платформы с полным обеспечением поддержки туманных вычислений (собственно вычисления, аналитика и организация транспортного уровня туманной сети), так и платформы, которые обеспечивают лишь транспортный уровень туманной сети и не обеспечивают управления вычислительными узлами и собственно туманными вычислениями на них. Платформы, обеспечивающие лишь транспортный уровень туманных вычислений, не будут рассматриваться в данной работе.

Можно выделить следующие ключевые характеристики частных и публичных коммерческих туманных платформ (см. табл. 2–3).

1. **Поддерживаемые аппаратные платформы** — платформа может работать с любым устройством, поддерживающим виртуализацию или контейнеризацию, или только с ограниченным списком устройств — через драйвера или фирменные устройства. Smartiply Fog, ThingWorx и Cisco IOx работают только с собственным аппаратным обеспечением.
2. **Базовая технология разработки** — на базе какой исполняемой среды создаются и запускаются приложения.
3. **Открытость коммуникационных протоколов и SDK** — есть ли ограничения на приложения, которые могут использоваться в тумане: требуется ли портирование приложений, или в принципе могут исполняться только приложения, написанные с использованием специальных поставляемых SDK, как, например, в случае с ThingWorx, чьи туманные приложения должны быть написаны с использованием фирменного SDK для запуска в тумане.
4. **Технология развертывания** — какая из технологий развертывания туманных узлов используется, если известно.
5. **Возможности интеграции** — имеется ли возможность интеграции с другими платформами, например, корпоративными решениями или публичными облаками.

6. **Подключение внешних источников данных** — возможность платформы подключаться к сторонним базам и хранилищам данных, физически расположенных вне центрального облака для хранения и обработки данных.
7. **Доступность дополнительных сервисов (Machine Learning, Analytics и т.п.)** — возможность подключения и использования дополнительно предоставляемых сервисов, которые предоставляют дополнительный функционал по анализу и работе с данными в тумане.
8. **Поддержка Edge** — возможность подключения и использования краевых устройств и краевых вычислений, и дальнейшего сбора и обработка информации от них.

4.1. Частные туманные платформы

Частные туманные платформы обеспечивают создание частных туманных решений на базе вычислительной инфраструктуры, развернутой непосредственно на ресурсах заказчика.

Платформа Cisco IOx была представлена компанией Cisco в 2014 году [2] как развитие сетевой инфраструктуры в связи с ожидаемым ростом IoT. Основной упор в платформе сделан на уменьшение трудозатрат по портированию приложений на туманные узлы, что достигается за счет использования технологий контейнеризации и базирования собственной ОС на основе Linux системы.

Cisco IOx — это среда приложений, которая объединяет в себе Cisco IOS (мини операционная система, устанавливаемая на всю аппаратуру Cisco) и Linux. Для разработки приложений применяются утилиты Linux с открытым исходным кодом. Используется единый протокол взаимодействия туманных приложений во всей сети, организованной с использованием технологий Cisco IoT. Туманные приложения, которые можно запускать на инфраструктуре IOx, поставляются как Cisco, так и партнерами компании. Разработку приложений можно вести на множестве языков программирования общего назначения.

Для разработки и развертывания приложений используется Docker. Поддерживаются разнообразные типы приложений, включая Docker-контейнеры и виртуальные машины (если такая возможность есть у сетевого оборудования). Также возможно использовать собственную исполняемую среду IOx для написания приложений на высокоуровневых языках программирования (например, Python).

Платформа Nebbiolo Technologies нацелена на корпоративный индустриальный рынок, который поддерживает концепцию Индустрии 4.0 [89]. Компания Nebbiolo Technologies тесно сотрудничает с Toshiba Digital Solutions [93] в поставке готовых вычислительных решений для промышленного и IoT сектора.

Платформа состоит из аппаратного обеспечения fogNode, программного стека fogOS и системного администратора fogSM, развертываемого в облаке или локально [32]. Fog System Manager (fogSM) предоставляет облачную платформу централизованного управления, которая позволяет развертывать и настраивать устройства на периферии.

Ключевой особенностью платформы является *fogOS* [32] — программный стек, обеспечивающий связь, управление данными и развертывание приложений на уровне тумана. Основанная на гипервизоре, fogOS предоставляет набор функций в виртуализированной форме. Поддерживается широкий спектр стандартов подключения устройств, а также позволяет размещать приложения и управлять ими в реальном времени.

Платформа ClearBlade представляет собой стек технологий, обеспечивающий быструю разработку и развертывание корпоративных IoT решений, начиная от краевых

устройств, заканчивая облачными сервисами. Она включает в себя программные компоненты, устанавливаемые на весь стек IoT устройств, а также обеспечивает возможность подключения сторонних систем через предоставляемый API для интеграции с устройствами, внутренними бизнес-приложениями и облачными сервисами. Платформа ClearBlade обеспечивает централизованную консоль управления IoT-приложениями, с возможностью развертывания как локально, так и в облаке. Функции управления платформой делегируются краевым узлам (либо на самих конечных устройствах или шлюзах к ним) при помощи системы туманных и краевых вычислений ClearBlade Edge [35].

Таблица 2

Обзор частных туманных платформ

Характеристика	ClearBlade	Smartiply Fog	LoopEdge	ThingWorx	Nebbiolo Technologies	Cisco Iox
Поддерживаемые аппаратные платформы	Универсально	Собственная аппаратура	Универсально	Собственная аппаратура	Универсально	Собственная аппаратура
Базовая технология разработки	JavaScript	Нет данных	Универсально (Docker)	Java VM	Универсально (Docker)	Docker, Linux, Iox
Открытость коммуникационных протоколов и SDK	+	+	+	–	+	+
Технология развертывания	Linux KVM	Нет данных	Docker	Нет данных	Docker	Linux KVM
Возможности интеграции	Oracle, SAP, Microsoft, Salesforce	–	–	Microsoft Azure IoT Hub	–	Microsoft Azure IoT Hub
Подключение внешних источников данных	+	–	+	+	+	+
Доступность дополнительных сервисов	Нет данных	+	–	+	+	+
Поддержка Edge	+	+	+	+	+	+

Платформой поддерживается бессерверный (англ. *Serverless computing*) подход к разработке сервисов на основе языка JavaScript, которые могут быть настроены на реализацию методов машинного обучения и анализа данных. Платформа обеспечивает механизмы экспорта данных и аналитики, собранной системой, в широко применяемые бизнес-системы, приложения и базы данных за счет интеграции с корпоративными платформенными решениями от Oracle, SAP, Microsoft и Salesforce. ClearBlade также предоставляет собственные панели управления, бизнес-приложения и системы управления базами данных для комплексного наблюдения и управления IoT экосистемой.

ClearBlade использует модель OAuth для контроля доступа, где каждый пользователь и устройство получает токен, который должен быть авторизован для получения доступа в систему или ее узлу. Данные шифруются как на самих устройствах, так и пе-

редачи по сети. Передаваемые данные шифруются с помощью библиотек OpenSSL с TLS-шифрованием.

Платформа Smartiply Fog — это туманная вычислительная платформа, которая делает упор на оптимизацию ресурсов и поддержание работы устройств в системе даже без подключения к облаку. Для онлайн сред платформа обеспечивает более высокую надежность благодаря оптимизации ресурсов и вычислений, которые производятся на базе аппаратуры собственного производства [85]. Платформа обеспечивает взаимодействие между устройствами по принципу точка-точка. Таким образом, система узлов может продолжать автономную работу получения, анализа и хранения данных, вплоть до восстановления связи с внешней сетью [88].

Платформа LoopEdge от Litmus Automation позволяет подключать различные устройства в единую систему, собирать и анализировать данные от них. Также Litmus Automation предоставляет отдельную платформу Loop, позволяющую управлять жизненным циклом любого IoT устройства и экспортировать данные в реальном времени во внутренние аналитические и бизнес-приложениями.

Разработчики платформы подчеркивают, что она способна работать с практически любым устройством, причем как с промышленным, так и доступным бытовому потребителю. Например, платформа поддерживает подключение устройств на базе Arduino и Raspberry Pi. Даже если какое-то устройство не поддерживается, подключить его к платформе достаточно легко, за счет устанавливаемых на само устройство исполняемых пакетов, которые можно расширять и создавать с нуля под конкретное устройство. Все инструкции находятся в открытом доступе.

Данная платформа получила широкое распространение среди известных машиностроительных концернов: Nissan, Renault, Mitsubishi Corporation Techno.

Платформа PTC ThingWorx — это IoT платформа, предлагающая подключение достаточно большого числа устройств (доступны драйвера для 150 видов устройств). Однако из-за того, что подключение устройств осуществляется через драйвера, которые требуют установки, прежде чем устройство можно использовать в тумане, эта платформа не является универсальной и имеет ограничения по используемым устройствам.

Приложения для платформы требуется писать с используемых поставляемых SDK. Дальнейший анализ данных и управление бизнес-процессами идет также через предоставляемые инструменты самой платформы. Для выполнения этих задач платформа имеет обширный раздел для разработчиков с инструкциями и учебными пособиями, а также помощью специалистов от самой компании для установки, настройки и расширения платформы. Также «из коробки» имеется возможность подключения к Microsoft Azure IoT Hub.

4.2. Публичные туманные платформы

Публичные туманные платформы сегодня представляют собой решения крупных игроков на рынке облачных вычислений, ориентированные на решения задач обработки данных от IoT-систем, привязанные к возможностям соответствующей облачной платформы. Ключевые характеристики рассматриваемых публичных туманных платформа приведены в табл. 3.

Платформа Azure IoT предоставляет платформу для туманных и краевых вычислений на основе стека технологий от компании Microsoft. Платформа Azure IoT состоит из нескольких крупных подсистем, таких как IoT Central, IoT Edge, которые в свою очередь основывают свою работу на облачной технологии Microsoft Azure. Подключение устройств от партнеров Microsoft возможно без использования драйверов или

программного кода за счет технологии *IoT Plug and Play*. Такой подход возможен для устройств под управлением любой ОС, включая Linux, Android, Azure Sphere OS, Windows IoT, RTOS и другие.

Создание, установка и управление туманными приложениями осуществляется через портал *Azure IoT Hub*. IoT Hub — это управляемая служба, размещенная в облаке, которая выступает в качестве центрального обработчика сообщений для двунаправленной связи между IoT приложением и устройствами, которыми оно управляет. IoT Hub поддерживает передачу данных как от устройства к облаку, так и от облака к устройству. IoT Hub поддерживает несколько шаблонов обмена сообщениями, таких как телеметрия между устройствами и облаками, загрузка файлов с устройств и технологию «запрос-ответ» для управления устройствами из облака.

Для развертывания вычислений ближе к самим устройствам или же на самих устройствах использует *Azure IoT Edge*, которая позволяет разворачивать приложения с собственной бизнес логикой или уже имеющиеся в каталоге готовые приложения на конечных устройствах при помощи технологии контейнеризации.

Платформа Amazon AWS IoT Greengrass позволяет распространить возможности AWS (Amazon Web Services) на периферийные устройства, что позволяет им локально работать с данными, используя при этом облако для управления, анализа и надежного хранения данных. AWS IoT Greengrass позволяет подключенным устройствам выполнять функции AWS Lambda, запускать контейнеры Docker, формировать прогнозы на основе моделей машинного обучения, синхронизировать данные устройств и безопасно взаимодействовать с другими устройствами даже без подключения к Интернету.

AWS IoT Greengrass позволяет создавать решения IoT, которые подключают различные типы устройств к облаку и друг к другу. AWS IoT Greengrass Core можно использовать на устройствах под управлением Linux (в том числе дистрибутивов Ubuntu и Raspbian), которые поддерживают архитектуры Arm или x86. Сервис AWS IoT Greengrass Core обеспечивает локальное исполнение кода AWS Lambda, передачу сообщений, управление данными и безопасность. Устройства с AWS IoT Greengrass Core выступают в качестве порталов сервиса и могут взаимодействовать с другими устройствами, на которых работает FreeRTOS (Real-time operating system for microcontrollers) или установлен пакет SDK AWS IoT для устройств. Размер таких устройств может быть очень разным: от небольших устройств на базе микроконтроллеров до крупных бытовых приборов. Когда устройство с AWS IoT Greengrass Core теряет связь с облаком, устройства в группе AWS IoT Greengrass могут продолжать взаимодействовать друг с другом по локальной сети.

Платформы Google, Yandex и Mail.ru предоставляют собственные облачные и туманные решения для сбора, хранения, обработки, анализа и визуализации данных. Собранные данные с устройств интегрируются в публичную облачную систему для более глубокой обработки и анализа (включая машинное обучение и искусственный интеллект) за счет высоких вычислительных мощностей облака. Данные платформы поддерживают множество протоколов подключения и взаимодействия через предоставляемое API. Имеется большое количество готовых к использованию сервисов, доступные для установки в каталоге самой платформы, которые можно подключать к собственному туманному решению, комбинируя между собой.

Таблица 3

Обзор публичных туманных платформ

Характеристика	AWS Green-grass	Azure IoT	Google	Yandex	Mail.ru
Поддерживаемые аппаратные платформы	Универсально	Универсально	Универсально	Универсально	Универсально
Базовая технология разработки	Универсально (Docker)	Универсально (Docker)	Универсально	Универсально	Универсально
Открытость коммуникационных протоколов и SDK	+	+	+	+	+
Технология развертывания	Docker	Docker	Docker	Docker	Docker
Возможность интеграции	Amazon Elastic Compute 2	Azure, через API	Сервисы Google и партнеров, через API	Универсально через API	Универсально через API
Подключение внешних источников данных	–	–	+	+	–
Доступность дополнительных сервисов (Machine Learning, Analytics и т.п.)	+	+	+	+	+
Поддержка Edge	+	+	+	+	+

4.3. Туманные платформы с открытым исходным кодом

В ходе анализа существующих решений нами был проведен обзор существующих туманных платформ с открытым исходным кодом. В отличие от коммерческих решений, для open source платформ представлены полные описания архитектур, требований к вычислительным ресурсам, а также используемым технологиям, как на аппаратном, так и программном уровнях (см. табл. 4).

Таблица 4

Обзор туманных платформ с открытым исходным кодом

	Цель	Внедрение
FogFrame2.0	Проверить концептуальную модель	–
FogFlow	Более простая и гибкая оркестрация сервисов	+
FogBus	Преодолеть неоднородность на уровне связи ОС и P2P различных узлов тумана	–

Платформа FogFrame2.0 — туманная платформа с открытым исходным кодом [91], нацеленная на развертывание на одноплатных компьютерах (Raspberry Pi). Была представлена для решения следующих задач [63]:

- определить и реализовать функции туманной инфраструктуры, т.е. создание и поддержание туманного ландшафта (множества туманных узлов и устройств) и управление приложениями [64, 65];
- реализовать эвристические алгоритмы для размещения услуг в тумане, а именно алгоритм первичной подгонки и генетический алгоритм;
- ввести механизмы адаптации к динамическим изменениям в тумане и для восстановления после перегрузок и сбоев.

Чтобы оценить поведение FogFrame, применялись различные модели поступления запросов приложений (константы, пирамиды и случайные обходы), а также исследовались процессы размещения вычислительных сервисов. Целью исследования в итоге было наблюдение за поведением платформы и ее реакцией на сбой. Платформа динамически реагирует на события во время выполнения, то есть, когда новые устройства появляются или отключаются, когда устройства испытывают сбой или перегрузки, выполняются необходимые передислокации узлов.

Платформа FogFlow — это туманная платформа с открытым исходным кодом [90]. Основной задачей разработчиков данной платформы было обеспечение простого и гибкого способа разработки, развертывания и оркестрации туманных сервисов [8]. Уникальность их подхода заключается в:

- контекстно-зависимой оркестрации сервисов в то время, как другие сервисы оперируют просто событиями (event) или группами событий (topic), происходящими в сети;
- сервисы и приложения FogFlow разработаны с учетом общего представления всех облачных узлов и пограничных узлов, а не с точки зрения каждого отдельного пограничного узла.

FogFlow предлагает использовать программную модель потоков данных, дополненную декларативными подсказками, основанными на широко используемом стандарте NGSI, что дает разработчикам услуг два преимущества:

- 1) быстрая и простая разработка приложений туманных вычислений, т.к. подобные подсказки скрывают множество настроек и сложностей развертывания от разработчиков сервисов;
- 2) высокая степень открытости и функциональная совместимость для обмена информацией и интеграции источников данных.

FogFlow является одним из компонентов большой открытой инфраструктуры IWARE [19], обеспечивающей разработку и внедрение различных smart решений [7, 10, 17]. Эта инфраструктура является одним из современных облачных фреймворков наряду с Amazon Web Services [25]. Для внедрения и использования FogFlow доступна широкая библиотека готовых решений от сообщества разработчиков и подробные инструкции по внедрению [87].

Платформа FogBus (при поддержке Melbourne Clouds Lab) объединяет различные аппаратные инструменты через программные компоненты, которые обеспечивают структурированное взаимодействие и независимое от платформы выполнение приложений [71]. FogBus применяет блокчейн для обеспечения целостности данных при передаче конфиденциальных данных. Независимая от платформы архитектура исполнения приложений и взаимодействия между узлами позволяет преодолеть неоднородность в интегрированной среде.

FogBus поддерживает реализацию различных политик управления ресурсами и планирования для выполнения приложений IoT, составленных с использованием моделей параллельного программирования, таких как SPMD (single program, multiple data).

Для оценки характеристик платформы FogBus используется прототип прикладной системы для анализа данных Sleep Apnea. На этом примере иллюстрируется, как приложение (в области здравоохранения), составленное с использованием модели SPMD может быть реализовано с использованием различных настроек FogBus для обработки IoT-данных в интегрированной вычислительной среде.

Данный фреймворк облегчает развертывание приложений IoT, мониторинг ресурсов и управление ими. Системные сервисы FogBus разработаны на кроссплатформенных языках программирования (PHP и Java) и используются с расширяемым протоколом прикладного уровня (HTTP), который помогает FogBus преодолевать неоднородность на уровне связи ОС и P2P различных узлов тумана. Кроме того, платформа FogBus функционирует как модель «Платформа как услуга» (PaaS) для интегрированной среды Fog Cloud, которая не только помогает разработчикам приложений создавать различные типы приложений IoT, но также поддерживает пользователей для настройки служб, и поставщики услуг для управления ресурсами в соответствии с условиями системы.

4.4. Методы классификации туманных платформ

Для формирования единого подхода к классификации туманных платформ, нами были рассмотрены ключевые туманные платформы и их ключевые характеристики. Так, например, **AWS Greengrass** способна работать без доступа к публичному облаку¹, однако в таком режиме работы возможно лишь хранение локальных данных. Центральное управление устройствами, а также централизованный сбор и обработка данных становится невозможен. Для полноценного функционирования платформы требуется доступ до AWS IoT Core, выступающий центральным сервисом для управления и организации работы тумана и который является публичным облаком.

Azure IoT также может работать в частных сетях², но при условии, что внутри частной сети будет шлюз, который должен подключаться к центральному узлу управления и сбора данных, и данный узел также является публичным облаком. Отличие от обычной организации работы тумана с публичным облаком заключается в наличии единой точки выхода во внешнюю сеть, а не множества различных шлюзов, которые общаются с публичным облаком.

Другие публичные туманные платформы имеют те же ограничения в отличие от частных туманных платформ и платформ с открытым исходным кодом, центральный узел управления которых может быть развернут на любом сервере в локальной сети или отсутствовать вовсе (в таком случае задачи управления и оркестрации разделяются между промежуточными узлами тумана, как это сделано, например, у FogFlow2.0).

Поэтому все туманные платформы можно классифицировать **по признаку открытости или закрытости развертывания хаба (Hub)** — сервиса, который отвечает за подключение, мониторинг и управление подключенными в туман устройствами. В том или ином виде хаб имеют практически все коммерческие туманные платформы: LoopEdge и Azure IoT так и называют данный сервис — Hub. Платформы ClearBlade и

¹https://aws.amazon.com/ru/greengrass/faqs/#Local_Resource_Access

²<https://azure.microsoft.com/en-us/blog/introducing-iot-hub-device-streams-in-public-preview/>

FogHorn имеют сервис с тем же функционалом, но называется он — Device Manager. У AWS Greengrass этот сервис называется AWS IoT Core.

Однако управление устройствами имеют только те системы, которые имеют Edge подсистемы. В остальных случаях туман оперирует не устройствами, а развернутыми вычислительными узлами, например, AWS IoT Greengrass Core¹, которые разворачиваются на устройстве, где это возможно, но о самом устройстве ничего неизвестно, взаимодействие идет с виртуализированным узлом на этом устройстве.

Также в предыдущих разделах рассматривались платформы, которые в некоторых случаях могли быть развернуты только на определенном перечне устройств, поставляемых самой платформой. Другие же туманные платформы не имели данного ограничения и могли работать с оборудованием пользователя (если оно удовлетворяет требованиям развертывания платформы). Это показатель классификации туманных платформ **по признаку открытости или закрытости аппаратной инфраструктуры**.

Тот же принцип наблюдается и при сравнении платформ **по признаку открытости или закрытости программной инфраструктуры**: платформа может поддерживать открытые протоколы обмена данными между узлами тумана или же туманные программы поставляются исключительно разработчиками самой платформы и лицензированными партнерами.

Таким образом, любую туманную платформу можно классифицировать по принципу открытости или закрытости ее компонентов (см. рис. 2). Также стоит отметить, что платформы с публичным хабом в большей степени стремятся к открытости своих аппаратных и программных инфраструктур.

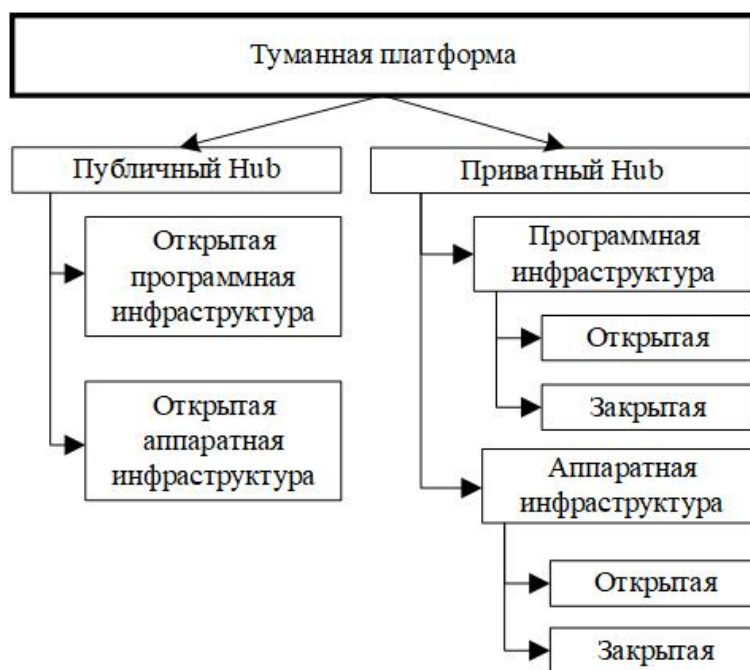


Рис. 2. Классификация туманных платформ по принципу открытости или закрытости ее компонентов

¹AWS IoT Core — это центральная точка управления туманом. AWS IoT Greengrass Core — это виртуализируемая система, разворачиваемая на устройстве, которое должно выступать вычислительным узлом.

Помимо открытости или закрытости своих компонентов некоторые платформы делали упор на доступность различных предоставляемых возможностей или сервисов, которыми обладает платформа. **Azure IoT Hub**, который является неотъемлемой частью платформы Azure IoT, явно называет себя **PaaS** (Platform as a Service), предоставляя готовые решения для реализации требуемых задач пользователя. Стоит отметить, что ни одна из публичных туманных платформ не позиционирует свои платформы как чисто туманные. Они предоставляют туманные вычисления как некий базовый функционал, который лежит в основе остальных предоставляемых функций и сервисов платформы.

Таким образом, сами платформы позиционируют некоторый функционал как базовый, который должен быть в любой туманной платформе, а пользователь заинтересован уже не только в простом развертывании и базовом управлении туманными узлами, а в решении своих конкретных задач: Индустрия 4.0, медицина, умный город и т.п. Платформы должны максимально предоставить готовые решения для каждой из задач пользователя.

Помимо прочего, некоторые платформы позволили пользователям обмениваться собственными готовыми решениями, созданными в рамках платформы при помощи «магазинов» — ресурсов, где пользователь может опубликовать свое готовое туманное приложение. Это привело к появлению целых туманных экосистем — **EaaS** (Ecosystem as a Service), которые позволяют создать собственное туманное решение из готовых компонентов, доступных на платформе.

Под данное описание также попадают и Open Source решения, которые предоставляют лишь базовый уровень функционала — **DaaS** (Deploy as a Service): разворачивание туманных узлов на имеющихся устройствах, оркестрацию и т.п. С другой стороны, FogFlow имеет более широкий функционал и даже собственную экосистему, в которую включены готовые к установке компоненты как от разработчиков платформы¹, так и от сообщества.

Классификацию «as a service» можно привести как способ классификации на основе предоставляемого функционала платформы (см. рис. 3).

Заключение

Увеличение передаваемых объемов данных и повышенная нагрузка на облако для работы клиентских сервисов стали предпосылкой к появлению концепции туманных вычислений. В данной работе была рассмотрена концепция туманных вычислений, их определение и ключевые характеристики. Также были рассмотрены, классифицированы и обобщены некоторые туманные платформы, которые являются предметами исследования или уже используются бизнесом и частными клиентами. В конце были описаны общие архитектурные характеристики присущие всем рассмотренным платформам.

Туманные вычисления являются более гибким и эффективным видом вычислений по сравнению с облачными за счет решения задач, требующих высокой пропускной способности вычислительной сети, возможности работы с географически рассредоточенными источниками данных, сверхнизкими задержками и обеспечением локальности обработки данных.

¹<https://fogflow.readthedocs.io/en/latest/setup.html>

EaaS (Ecosystem as a Service)	Публичное облако	Магазин приложений	
PaaS (Platform as a Service)	Machine Learning	Stream Processing	Другие дополнительно предоставляемые сервисы
	Аналитика	Поддержка Edge	Подключение к Enterprise сервисам
DaaS (Deploy as a Service)	Развертывание на туманных узлах	Оркестрация	Наличие готовых к установке компонентов
	Горизонтальное масштабирование	Приватное развертывание облака для тумана	

Рис. 3. Классификация туманных платформ на основе предоставляемого функционала платформы

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта No 18-07-01224 а и при финансовой поддержке Министерства науки и высшего образования РФ (государственное задание FENU-2020-0022).

Литература

1. Al-Doghman F., Chaczko Z., Ajayan A.R., et al. A review on Fog Computing Technology // Conference Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 (Budapest, Hungary, October, 9–12, 2017). Institute of Electrical and Electronics Engineers Inc., 2017. P. 1525–1530. DOI: 10.1109/SMC.2016.7844455.
2. Antonio S. Cisco Delivers Vision of Fog Computing to Accelerate Value from Billions of Connected Devices 2014. P. 1–4.
3. Armbrust M., Fox A., Griffith R., et al. A View of Cloud Computing // Communications of the ACM. 2010. Vol. 53, no. 4. P. 50–58. DOI: 10.1145/1721654.1721672.
4. Bonomi F., Milito R., Zhu J., et al. Fog Computing and Its Role in the Internet of Things // Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC'12 (Helsinki, Finland, August, 17, 2012). ACM Press, 2012. P. 13–15. DOI: 10.1145/2342509.2342513.
5. Brito M.S.D., Hoque S., Magedanz T., et al. A Service Orchestration Architecture for Fog-enabled Infrastructures // 2017 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017 (Valencia, Spain, May, 8–11, 2017). Institute of Electrical and Electronics Engineers Inc., 2017. P. 127–132. DOI: 10.1109/FMEC.2017.7946419.
6. Brynjolfsson E., Hofmann P., Jordan J. Cloud Computing and Electricity: Beyond the Utility Model // Communications of the ACM. 2010. Vol. 53, no. 5. P. 32–34. DOI: 10.1145/1735223.1735234.

7. Celesti A., Fazio M., Márquez F.G., et al. How to Develop IoT Cloud E-Health Systems Based on Fiware: A Lesson Learnt // Journal of Sensor and Actuator Networks. 2019. Vol. 8, no. 1. DOI: 10.3390/jsan8010007.
8. Cheng B., Solmaz G., Cirillo F., et al. FogFlow: Easy Programming of IoT Services Over Cloud and Edges for Smart Cities // IEEE Internet of Things Journal. 2018. Vol. 5, no. 2. P. 696–707. DOI: 10.1109/JIOT.2017.2747214.
9. Chiang M., Ha S., Chih-Lin I., et al. Clarifying Fog Computing and Networking: 10 Questions and Answers // IEEE Communications Magazine. 2017. Vol. 5, no. 4. P. 18–20. DOI: 10.1109/MCOM.2017.7901470.
10. Dantas L., Cavalcante E., Batista T. A Development Environment for FIWARE-Based Internet of Things Applications // Proceedings of the 2019 Workshop on Middleware and Applications for the Internet of Things, Part of Middleware 2019 Conference, M4IoT 2019 (Davis CA, USA, December, 7–11, 2019). Association for Computing Machinery, Inc, 2019. P. 21–26. DOI: 10.1145/3366610.3368100.
11. Dar B.K., Shah M.A., Islam S.U., et al. Delay-Aware Accident Detection and Response System Using Fog Computing // IEEE Access. 2019. Vol. 7. P. 70975–70985. DOI: 10.1109/ACCESS.2019.2910862.
12. Donno M. De, Tange K., Dragoni N. Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog // IEEE Access. 2019. Vol. 7. P. 150936–150948. DOI: 10.1109/ACCESS.2019.2947652.
13. Eder M. Hypervisor-vs. Container-Based Virtualization // Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM). 2016. Vol. 1.
14. Emeras J., Varrette S., Bouvry P. Amazon Elastic Compute Cloud (EC2) vs. In-house HPC Platform: A Cost Analysis // IEEE International Conference on Cloud Computing, CLOUD (Honolulu, USA, June, 25–30, 2017). IEEE, 2017. P. 284–293. DOI: 10.1109/CLOUD.2016.44.
15. Eugene G. Cloud Computing Models. 2013.
16. Evans D. The Internet of Things: How the Next Evolution of the Internet is Changing Every Thing // CISCO white paper. 2011. Vol. 1. P. 1–11.
17. Fazio M., Celesti A., Marquez F.G., et al. Exploiting the FIWARE Cloud Platform to Develop a Remote Patient Monitoring System // Proceedings of IEEE Symposium on Computers and Communications, 2016. Institute of Electrical and Electronics Engineers Inc., 2016. P. 264–270. DOI: 10.1109/ISCC.2015.7405526.
18. Feeney G.J. Utility computing — A Superior Alternative? // AFIPS Conference Proceedings — 1974 National Computer Conference, AFIPS 1974 (Chicago, Illinois, USA, May, 6–10, 1974). ACM Press, 1974. P. 1003–1004. DOI: 10.1145/1500175.1500370.
19. FIWARE "About us". 2015. URL: <https://www.fiware.org/about-us/> (дата обращения: 03.03.2020).
20. Foster I., Kesselman C. The History of the Grid // Advances in Parallel Computing. 2011. Vol. 20. P. 3–30. DOI: 10.3233/978-1-60750-803-8-3.
21. Garcia J., Simo E., Masip-Bruin X., et al. Do We Really Need Cloud? Estimating the Fog Computing Capacities in the City of Barcelona // Proceedings of the 11th IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion 2018 (Zurich, Switzerland, December, 17–20, 2019). 2019. P. 290–295. DOI: 10.1109/UCC-Companion.2018.00070.
22. GE Digital What is Edge Computing? | GE Digital. 2018. URL: <https://www.ge.com/digital/blog/what-edge-computing> (дата обращения: 03.03.2020).

23. Gu L., Zeng D., Guo S., et al. Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System // IEEE Transactions on Emerging Topics in Computing. 2017. Vol. 5, no. 1. P. 108–119. DOI: 10.1109/TETC.2015.2508382.
24. Guharoy R., Sur S., Rakshit S., et al. A Theoretical and Detail Approach on Grid Computing. A Review on Grid Computing Applications // 2017 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON 2017 (Bangkok, Thailand, August, 16–18, 2017). Institute of Electrical and Electronics Engineers Inc., 2017. P. 142–146. DOI: 10.1109/IEMECON.2017.8079578.
25. Guth J., Breitenbucher U., Falkenthal M., et al. Comparison of IoT Platform Architectures: A Field Study Based on a Reference Architecture // 2016 Cloudification of the Internet of Things, CIoT 2016, 2017. Institute of Electrical and Electronics Engineers Inc., 2017. DOI: 10.1109/CIOT.2016.7872918.
26. Hagi A., Wright J. When Data Creates Competitive Advantage...And When It Doesn't // Harvard Business Review. 2020. Vol. 98, no. 1. P. 94–101.
27. Hannabuss S. The Big Switch: Rewiring the World, from Edison to Google // Library Review. 2009. Vol. 58, no. 2. P. 136–137.
28. Haouari F., Faraj R., Alja'Am J.M. Fog Computing Potentials, Applications, and Challenges // 2018 International Conference on Computer and Applications, ICCA 2018 (Beirut, Lebanon, July, 25–26, 2018). IEEE, 2018. P. 399–406. DOI: 10.1109/COMAPP.2018.8460182.
29. Hashemi S.M., Bardsiri A.K. Cloud Computing vs. Grid Computing // ARPN Journal of Systems and Software. 2012. Vol. 2, no. 5.
30. Hilbrich M., Frank M. Abstract Fog in the Bottle - Trends of Computing in History and Future // Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2018 (Prague, Czech Republic, August, 29–31, 2018). Institute of Electrical and Electronics Engineers Inc., 2018. P. 519–522. DOI: 10.1109/SEAA.2018.00089.
31. Hofmann P., Woods D. Cloud Computing: The Limits of Public Clouds for Business Applications // IEEE Internet Computing. 2010. Vol. 14, no. 6. P. 90–93. DOI: 10.1109/MIC.2010.136.
32. Hong C.H., Varghese B. Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms // ACM Computing Surveys. 2019. Vol. 52, no. 5. P. 1–37. DOI: 10.1145/3326066.
33. Hong H.J. From Cloud Computing to Fog Computing: Unleash the Power of Edge and End Devices // Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom (Hong Kong, Hong Kong, December, 11–14, 2017). IEEE Computer Society, 2017. P. 331–334. DOI: 10.1109/CloudCom.2017.53.
34. Huang C., Lu R., Choo K.K.R. Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges // IEEE Communications Magazine. 2017. Vol. 55, no. 11. P. 105–111. DOI: 10.1109/MCOM.2017.1700322.
35. Hughes I., Immerman D., Daly P. ClearBlade Demonstrates Scalability and Edge Analytics With IoT Platform, 2017.
36. Iorga M., Feldman L., Barton R., et al. Fog Computing Conceptual Model. Gaithersburg, MD, 2018.
37. Jalali F., Hinton K., Ayre R., et al. Fog Computing May Help to Save Energy in Cloud Computing // IEEE Journal on Selected Areas in Communications. 2016. Vol. 34, no. 5. P. 1728–1739. DOI: 10.1109/JSAC.2016.2545559.

38. Jiang Y., Huang Z., Tsang D.H.K.K. Challenges and Solutions in Fog Computing Orchestration // IEEE Network. 2018. Vol. 32, no. 3. P. 122–129. DOI: 10.1109/MNET.2017.1700271.
39. Kakakhel S.R.U., Mukkala L., Westerlund T., et al. Virtualization at the Network Edge: A Technology Perspective // 2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018 (Barcelona, Spain, April, 23–26, 2018). Institute of Electrical and Electronics Engineers Inc., 2018. P. 87–92. DOI: 10.1109/FMEC.2018.8364049.
40. Kokkinou A., Cranage D.A. Using Self-Service Technology to Reduce Customer Waiting Times // International Journal of Hospitality Management. 2013. Vol. 33, no. 1. P. 435–445. DOI: 10.1016/j.ijhm.2012.11.003.
41. Kumar R., Charu S. Comparison Between Cloud Computing, Grid Computing, Cluster Computing and Virtualization // International Journal of Modern Computer Science and Applications. 2015. Vol. 8, no. 31. P. 2321–2632. DOI: 10.13140/2.1.1759.7765.
42. Lee J. A View of Cloud Computing // International Journal of Networked and Distributed Computing. 2013. Vol. 1, no. 1. P. 2–8. DOI: 10.2991/ijndc.2013.1.1.2.
43. Li C., Xue Y., Wang J., et al. Edge-Oriented Computing Paradigms: A Survey on Architecture Design and System Management // ACM Computing Surveys. 2018. Vol. 51, no. 2. DOI: 10.1145/3154815.
44. Liu L., Wang Y., Yang Y., et al. Utility-Based Computing Model for Grid // Proceedings of the 1st International Conference on Semantics, Knowledge and Grid, SKG 2005 (Beijing, China, November, 27–29, 2005). 2005. P. 109–109. DOI: 10.1109/SKG.2005.140.
45. Liu Y., Fieldsend J.E., Min G. A Framework of Fog Computing: Architecture, Challenges, and Optimization // IEEE Access. 2017. Vol. 5. P. 25445–25454. DOI: 10.1109/ACCESS.2017.2766923.
46. Madsen H., Albeanu G., Burtschy B., et al. Reliability in the Utility Computing Era: Towards Reliable Fog Computing // International Conference on Systems, Signals, and Image Processing (Rio de Janeiro, Brazil, June, 3–5, 2013). IEEE Computer Society, 2013. P. 43–46. DOI: 10.1109/IWSSIP.2013.6623445.
47. Mahmood Z., Ramachandran M. Fog Computing: Concepts, Principles and Related Paradigms // Springer International Publishing, 2018. P. 3–21. DOI: 10.1007/978-3-319-94890-4_1.
48. Mahmoudi C., Mourlin F., Battou A. Formal Definition of Edge Computing: An Emphasis on Mobile Cloud and IoT Composition // 2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018 (Barcelona, Spain, April, 23–26, 2018). Institute of Electrical and Electronics Engineers Inc., 2018. P. 34–42. DOI: 10.1109/FMEC.2018.8364042.
49. McAfee A., Brynjolfsson E. Big Data: The Management Revolution // Harvard Business Review. 2012. Vol. 90, no. 10. P. 4.
50. Mell P., Grance T. The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology. 2012. P. 97–101.
51. Naha R.K., Garg S., Georgakopoulos D., et al. Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions // IEEE Access. 2018. Vol. 6. P. 47980–48009. DOI: 10.1109/ACCESS.2018.2866491.
52. Nakagawa M., Hasegawa H., Sato K., et al. Adaptive Self-Reconfigurable Network to Create Cost-Effective Bandwidth-on-Demand Services // Optics InfoBase Conference Papers (Rochester, NY, United States, October, 24–28, 2010). 2010. DOI: 10.1364/nfoec.2010.nwa2.

53. Narula S., Jain A. Prachi Cloud Computing Security: Amazon Web Service // International Conference on Advanced Computing and Communication Technologies, ACCT (Rohtak, Haryana, India, February, 21–22, 2015). Institute of Electrical and Electronics Engineers Inc., 2015. P. 501–505. DOI: 10.1109/ACCT.2015.20.
54. OpenFog Consortium Architecture Working Group OpenFog Reference Architecture for Fog Computing. 2017.
55. Pinchuk A., Sokolov N., Freinkman V. General principles of foggy computing // Last-Mile. 2018. no. 3. P. 38–45. DOI: 10.22184/2070-8963.2018.72.3.38.45.
56. Proferansov D.Y., Safonova I.E. To the Question of Fog Computing and the Internet of Things // Educational Resources and Technology. 2017. Vol. 4, no. 21. P. 30–39.
57. Puliafito C., Mingozzi E., Vallati C., et al. Virtualization and Migration at the Network Edge: An Overview // Proceedings of 2018 IEEE International Conference on Smart Computing, SMARTCOMP 2018 (Taormina, Sicily, Italy, June, 18–20, 2018). Institute of Electrical and Electronics Engineers Inc., 2018. P. 368–374. DOI: 10.1109/SMARTCOMP.2018.00031.
58. Radchenko G.I., Alaasam A.B.A., Tchernykh A.N. Comparative Analysis of Virtualization Methods in Big Data Processing // Supercomputing Frontiers and Innovations. 2019. Vol. 6, no. 1. P. 48–79. DOI: 10.14529/jsfi190107.
59. Ravandi B., Papapanagiotou I. A Self-Learning Scheduling in Cloud Software Defined Block Storage // IEEE International Conference on Cloud Computing, CLOUD (Honolulu, Hawaii, USA, June, 25–July, 1, 2017). IEEE Computer Society, 2017. P. 415–422. DOI: 10.1109/CLOUD.2017.60.
60. Reinsel D., Gantz J. Extracting Value from Chaos // IDC Report. 2011. Vol. 1142. P. 1–12.
61. Sadashiv N., Kumar S.M.D. Cluster, Grid and Cloud Computing: A Detailed Comparison // Final Program and Proceedings of the 6th International Conference on Computer Science and Education, ICCSE 2011 (Chennai, India, December, 14–15, 2011). IEEE, 2011. P. 477–482. DOI: 10.1109/ICCSE.2011.6028683.
62. Sehgal N.K., Bhatt P.C.P., Sehgal N.K., et al. Features of Private and Public Clouds Cham: Springer International Publishing, 2018. P. 51–60.
63. Skarlat O., Karagiannis V., Rausch T., et al. A Framework for Optimization, Service Placement, and Runtime Operation in the Fog // Proceedings of the 11th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2018 (Zurich, Switzerland, December, 17–20, 2019). Institute of Electrical and Electronics Engineers Inc., 2019. P. 164–173. DOI: 10.1109/UCC.2018.00025.
64. Skarlat O., Nardelli M., Schulte S., et al. Optimized IoT Service Placement in the Fog // Service Oriented Computing and Applications. 2017. Vol. 11, no. 4. P. 427–443. DOI: 10.1007/s11761-017-0219-8.
65. Skarlat O., Schulte S., Borkowski M., et al. Resource Provisioning for IoT Services in the Fog // Proceedings of 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications, SOCA 2016 (Macau, China, November, 4–6, 2016). Institute of Electrical and Electronics Engineers Inc., 2016. P. 32–39. DOI: 10.1109/SOCA.2016.10.
66. Smirnov Y. Cloud computing. The History and Impact on Libraries' Future // Scientific and Technical Libraries. 2016. no. 6. P. 62–73. DOI: DOI: 10.33186/1027-3689-2016-6-62-73.

67. Sotomayor B., Montero R.S., Llorente I.M., et al. Virtual Infrastructure Management in Private and Hybrid Clouds // *IEEE Internet Computing*. 2009. Vol. 13, no. 5. P. 14–22. DOI: 10.1109/MIC.2009.119.
68. Taleb T., Samdanis K., Mada B., et al. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration // *IEEE Communications Surveys and Tutorials*. 2017. Vol. 19, no. 3. P. 1657–1681. DOI: 10.1109/COMST.2017.2705720.
69. Tseng F.H., Tsai M.S., Tseng C.W., et al. A Lightweight Autoscaling Mechanism for Fog Computing in Industrial Applications // *IEEE Transactions on Industrial Informatics*. 2018. Vol. 14, no. 10. P. 4529–4537. DOI: 10.1109/TII.2018.2799230.
70. Tuli S., Basumatary N., Buyya R. EdgeLens: Deep Learning Based Object Detection in Integrated IoT, Fog and Cloud Computing Environments // *Proceedings of the 4th IEEE International Conference on Information Systems and Computer Networks, ISCON 2019 (Mathura, India, November 21–22, 2019)*. IEEE Press, USA, 2019. P. 496–502. DOI: 10.1109/ISCON47742.2019.9036216.
71. Tuli S., Mahmud R., Tuli S., et al. FogBus: A Blockchain-Based Lightweight Framework for Edge and Fog Computing // *Journal of Systems and Software*. 2019. Vol. 154. P. 22–36. DOI: 10.1016/j.jss.2019.04.050.
72. Vandenberg A. *Grid Computing for All* // Charles River Media, 2005. P. 3. DOI: 10.1145/1167350.1167353.
73. Velasquez K., Abreu D.P., Assis M.R.M., et al. Fog Orchestration for the Internet of Everything: State-of-the-Art and Research Challenges // *Journal of Internet Services and Applications*. 2018. Vol. 9, no. 1. DOI: 10.1186/s13174-018-0086-3.
74. Wadhwa H., Aron R. Fog Computing with the Integration of Internet of Things: Architecture, Applications and Future Directions // *Proceedings of the 16th IEEE International Symposium on Parallel and Distributed Processing with Applications, 17th IEEE International Conference on Ubiquitous Computing and Communications, 8th IEEE International Conference on Big Data and Cloud Computing, (Melbourne, Australia, December, 11–13, 2019)*. IEEE, 2019. P. 987–994. DOI: 10.1109/BDCloud.2018.00144.
75. Webb K. Reviews. Architects of the Information Society: 35 Years of the Laboratory for Computer Science at MIT // *Internet Research*. 2000. Vol. 10, no. 1. P. 169–174.
76. Weinhardt C., Anandasivam A., Blau B., et al. Cloud Computing — A Classification, Business Models, and Research Directions // *Business & Information Systems Engineering*. 2009. Vol. 1, no. 5. P. 391–399. DOI: 10.1007/s12599-009-0071-2.
77. Wen Z., Yang R., Garraghan P., et al. Fog Orchestration for Internet of Things Services // *IEEE Internet Computing*. 2017. Vol. 21, no. 2. P. 16–24. DOI: 10.1109/MIC.2017.36.
78. Wood T., Ramakrishnan K.K., Shenoy P., et al. CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines // *IEEE/ACM Transactions on Networking*. 2015. Vol. 23, no. 5. P. 1568–1583. DOI: 10.1109/TNET.2014.2343945.
79. Yang J., Pang J., Qi N., et al. On-Demand Self-Adaptivity of Service Availability for Cloud Multi-Tier Applications // *Proceedings of 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015, 2015*. Institute of Electrical and Electronics Engineers Inc., 2015. P. 1237–1240. DOI: 10.1109/CCGrid.2015.146.
80. Yousefpour A., Fung C., Nguyen T., et al. All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey // *Journal of Systems Architecture*. 2019. Vol. 98. P. 289–330. DOI: 10.1016/j.sysarc.2019.02.009.

81. Zhang B., Mor N., Kolb J., et al. The Cloud is Not Enough: Saving IoT from the Cloud // 7th USENIX Workshop on Hot Topics in Storage and File Systems, HotStorage 2015, 2020.
82. Zhang P., Liu J. K., Richard Yu F., et al. A Survey on Access Control in Fog Computing // IEEE Communications Magazine. 2018. Vol. 56, no. 2. P. 144–149. DOI: 10.1109/MCOM.2018.1700333.
83. Zlatanov N. The Data Center Evolution from Mainframe to Cloud // IEEE Computer Society. 2016. DOI: 10.13140/RG.2.1.4103.8489.
84. A Guide to Edge IoT Analytics: Internet of Things Blog. URL: <https://www.ibm.com/blogs/internet-of-things/edge-iot-analytics/> (дата обращения: 02.03.2020).
85. Edge Gateway | Smartiply. URL: <https://www.smartiply.com/gateway> (дата обращения: 02.03.2020).
86. Fog computing brings new business opportunities and disruptions - IoT Agenda. URL: <https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Fog-computing-brings-new-business-opportunities-and-disruptions> (дата обращения: 27.02.2020).
87. FogFlow — FogFlow v2.0 documentation. URL: <https://fogflow.readthedocs.io/en/latest/> (дата обращения: 03.03.2020).
88. Mobile Platform | Smartiply. URL: <https://www.smartiply.com/mobile> (дата обращения: 02.03.2020).
89. Nebbiolo Technologies — Pioneers in Fog Computing. URL: <https://www.nebbiolo.tech/> (дата обращения: 02.03.2020).
90. Smartfog/Fogflow: FogFlow is a Standard-Based IoT Fog Computing Framework that Supports Serverless Computing and Edge Computing with Advanced Programming Models. URL: <https://github.com/smartfog/fogflow> (дата обращения: 03.03.2020).
91. Softls/FogFrame-2.0: FogFrame Framework (with Extensions). URL: <https://github.com/softls/FogFrame-2.0> (дата обращения: 02.03.2020).
92. The Industrial Internet Consortium and Openfog Consortium Join Forces | Industrial Internet Consortium. URL: <https://www.iiconsortium.org/press-room/01-31-19.htm> (дата обращения: 27.02.2020).
93. Toshiba Digital Solutions Corporation and Nebbiolo Technologies Inc. Sign an Industrial IoT Strategic Partnership Agreement. URL: <https://www.prnewswire.com/news-releases/toshiba-digital-solutions-corporation-and-nebbiolo-technologies-inc-sign-an-industrial-iot-strategic-partnership-agreement-300632595.html> (дата обращения: 02.03.2020).

Кирсанова Александра Александровна, научный сотрудник кафедры электронных вычислительных машин, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Радченко Глеб Игоревич, к.ф.-м.н., доцент, директор Высшей школы электроники и компьютерных наук, заведующий кафедрой электронных вычислительных машин, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Черных Андрей Николаевич, к.т.н., доцент, заведующий научно-исследовательской лабораторией проблемно-ориентированных облачных сред, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация), профессор, научно-исследовательский центр Энсенеды (Энсенета, Мексика)

OVERVIEW OF FOG COMPUTING ORGANIZATION TECHNOLOGIES

© 2020 A.A. Kirsanova¹, G.I. Radchenko¹, A.N. Chernykh^{1,2}

¹South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

²Ensenada Research Center

(Carretera Ensenada - Tijuana No. 3918, Ensenada, 22860 Mexico)

E-mail: alexander.a.kirsanov@susu.ru, gleb.radchenko@susu.ru, chernykh@cicese.mx

Received: 14.06.2020

As the Internet of Things (IoT) becomes a part of our daily life, there is a rapid growth in the number of connected devices. A well-established approach based on cloud computing technologies cannot provide the necessary quality of service in such an environment, particularly in terms of reducing data latency. Today, fog computing technology is seen as a promising solution for processing large amounts of critical and time-sensitive data. This article reviews cloud computing technology and analyzes the prerequisites for the evolution of this approach and the emergence of the concept of fog computing. As part of an overview of the key features of fog computing, we analyze the frequent confusion with the fusion of the concepts of fog and edge computing. The paper provides an overview of fog computing technologies: virtualization, containerization, and orchestration, as well as a systematic analysis of the most popular platforms that support fog computing. As a result of the analysis, we offer two approaches to the classification of fog-computing platforms: on the principle of openness/closure of components, as well as three-level classification based on the provided platform functionality (Deploy-, Platform- and Ecosystem as a Service).

Keywords: cloud computing, fog computing, edge computing, internet of things.

FOR CITATION

Kirsanova A.A., Radchenko G.I., Chernykh A.N. Overview of Fog Computing Organization Technologies. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2020. Vol. 9, no. 3. P. 35–63. (in Russian) DOI: 10.14529/cmse200303.

This paper is distributed under the terms of the Creative Commons Attribution—Non-Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.