Моделирование и анализ информационных систем. Т. 26, №1 (2019), с. 146–169
Modeling and Analysis of Information Systems. Vol. 26, No 1 (2019), pp. 146–169

DOI: 10.18255/1818-1015-2019-1-146-169

УДК 004.7

Иерархические периферийные вычисления

Смелянский Р. Л.

Поступила в редакцию 10 января 2019
После доработки 12 февраля 2019
Принята к публикации 14 февраля 2019

Аннотация. На смену вычислительной парадигме, основанной на giant-like ЦОДах, идет новая, основанная на сети мелких ЦОДов, образующих инфраструктуру для облачных вычислений. Эта смена объективна. Её актуальность обусловлена требованиями новых приложений, активно использующих видео, интерактивность в реальном времени, новые технологии мобильной связи, которые сегодня невозможно реализовать без облачных вычислений и виртуализации на основе технологий SDN&NFV. В статье рассмотрены требования, предъявляемые этими приложениями, предложена архитектура новой парадигмы, которую мы называем «Иерархическими периферийными вычислениями» (Hierarchical Edge Computing – HEC). Показано, что большинство современных приложений являются распределенными совокупностями сервисов реального времени, которые требуют гарантированного качества обслуживания и возможности динамически быть размещенными при работе на периферии сетей разных операторов. Обсуждаются основные научные проблемы, которые необходимо решить для реализации предлагаемой новой парадигмы.

Ключевые слова: ПКС, программно-конфигурируемые сети, ВСС, виртуализация сетевых сервисов, туманные вычисления, облачные вычисления, ЦОД, центры обработки данных, мобильные периферийные вычисления

Для цитирования: Смелянский Р. Л., "Иерархические периферийные вычисления", *Моделирование и анализ информационных систем*, **26**:1 (2019), 146–169.

Об авторах:
Смелянский Руслан Леонидович, чл.-кор. РАН, д-р физ.-мат. наук, проф., orcid.org/0000-0003-2311-4513
Московский государственный университет имени М.В. Ломоносова,
Ленинские горы, 1, стр. 52, г. Москва, 119991, Россия, e-mail: smel@cs.msu.ru

Благодарности:
Работа выполнена при поддержке РФФИ, грант N 18-07-01245.

Введение

Мы быстро движемся из эпохи «создай свое» в эпоху просто «используй услугу». От организации вычислений, когда предприятие вынуждено покупать себе все необходимое сетевое оборудование, подключение и услуги провайдера, нанимать дорогостоящих Cisco Certified Internetworking Experts (CCIE) специалистов, которые должны заставить все это заработать, – к организации вычислений, когда предприятие просто использует «сеть как услугу». В этой новой развивающейся парадигме «сеть

как услуга» можно прокладывать и убирать соединения, разворачивать услуги, тогда и там, где они нужны, и оплачивать только то, что было использовано.

За последние 10 лет облачные вычисления, как вычислительная парадигма, полностью изменили ландшафт информационно-коммуникационных технологий (ИКТ) [1]–[4]. Это значительно способствовало росту как числа центров обработки данных (ЦОД), так и их размера, увеличению пропускной способности магистральных каналов [2], увеличению плотности оборудования: виртуализация ИТ-оборудования в облачных архитектурах позволила втиснуть в одну стойку то, что раньше требовало 10 стоек. Совершенствование и развитие возможностей персональных гаджетов, различных типов датчиков, развитие технологий передачи данных, таких как OTN, сети 5G, сетевая конвергенция, появление технологий программно-конфигурируемых сетей (ПКС)[1] и виртуализации сетевых функций (ВСС)[2] дало толчок развитию большого числа приложений реального времени [5]. Вот лишь некоторые примеры таких приложений: умный город, умный дом, здравоохранение (особенно такие его области, как хирургия, телемедицина, экстренная кардиология), интерактивные игры, обучение, дополненная реальность, сельское хозяйство, инфраструктура для научных многодисциплинарных исследований, социальные коммуникации, системы управления объектами энергетики (умные сети электроснабжения), беспроводные датчики, встроенные в различные роботизированные устройства, мониторинг и управление транспортными системами и объектами, сборочные производственные линии, газопроводы и нефтепроводы. Согласно оценкам IDC, в 2019 году 49% всего трафика наших сетей будет генерироваться такими приложениями.

Все вышеупомянутые приложения чувствительны к задержкам на время реакции, требуют интеграции географически распределенных данных. Эти ограничения могут быть жесткими или нет, в зависимости от приложений. Они проистекают либо из характера самих приложений, либо из требований пользователей, которые готовы ждать ответа лишь ограниченное время. При этом качество представления ответа должно удовлетворять ожиданиям пользователя. Другими словами, все больше и больше наших приложений становятся приложениями реального времени.

Ограничения, которые налагают приложения как жёсткого, так и мягкого реального времени, имеют две основных составляющих: общее время на передачу исходных данных и результатов их обработки, а также собственно время их обработки. Баланс между этими составляющими существенно зависит от архитектуры вычислительной (обрабатывающей) и коммуникационной инфраструктуры (ИКТ-инфраструктуры).

Усиление ограничений на время взаимодействия между приложением и терминальным устройством пользователя привело к противоречию с концепцией вычислений, основанной на giant-like ЦОДах. Дело в том, что в инфраструктуре на основе giant-like ЦОДов наши возможности изменять задержку при передаче данных и их обработке весьма ограничены. В таблице 1 на основе данных из [30] показаны такие характеристики взаимодействия, как задержки, потери пакетов и пропускной способности в зависимости от расстояния между ЦОД и конечным пользователем.

В свое время одним из основных доводов за концепцию giant-like ЦОДов были экономические соображения снижения общей стоимости владения ИКТ-инфра-

[1]Англ. эквивалент Software Defined Network (SDN).
[2]Англ. эквивалент Network Function Virtualization (NFV)

Таблица 1. Характеристики взаимодействия клиента и сервера

Table 1. Interaction characteristics of a client and the server

Расстояние (между клиентом и сервером)	**RTT**	**Потери пакетов**	**Пропускная способность**	**Загрузка 4 GB**
<160 км	1.6 мс	0.6%	44 Mbps	12 мин
<750–1600 км	16 мс	0.7%	4 Mbps	2.2 ч
∼ 4000 км	48 мс	1.0%	1 Mbps	8.2 ч
∼ 9000 км	96 мс	1.4%	0.4 Mbps	20 ч

структурой такого ЦОД. Однако рост числа мобильных устройств (по данным аналитиков, 50 миллиардов штук к 2020 году, т.е. на следующий год) приводит к резкому росту объёма сгенерированных данных, которые необходимо обработать и на которые необходимо должным образом отреагировать [6]. Ярким примером тому является «Интернет вещей» (IoT). Миллиарды IoT-устройств будут наводнять наши сети зеттабайтами данных, которые требуют обработки и ответа в режиме реального времени. Согласно отчету Cisco Systems [7], к 2019 году 500 ZB данных будут «прокачиваться» через наши сети в год, а к 2020 году каждый день – 2,3 ZB. IoT-приложения требуют высокой степени мобильности, строго ограниченной задержки доставки данных и обработки их в режиме реального времени [8]. Для обработки такого потока данных в реальном времени требуются новые решения и новые подходы к организации вычислительной и коммуникационной инфраструктуры.

1. Вычисления на периферии сегодня

На сегодняшний день было предложено несколько подходов к организации ИКТ-инфраструктуры для приложений, для которых решающее значение имеет время отклика: туманные вычисления (Fog computing), облачка – мобильные облачные мини-ЦОДы (Cloudlets), мобильные вычисления на границе (MEC), микро-ЦОДы (MDC). Все они в основном сосредоточены на потребностях IoT и имеют ограниченную интерпретацию концепции периферии (Edge) сети. Например, это может быть зона доступа в сеть, расположенная как можно ближе к пользовательскому устройству или датчику (Customer Premise Equipment – CPE) и оснащенная средствами вычисления, хранения и накопления данных. Ниже кратко рассмотрены перечисленные выше подходы.

Fog computing представляет собой платформу, которая обеспечивает облачные вычисления в непосредственной близости от конечных пользователей. Первоначально термин «туман» был введен Cisco [9]: виртуализированная «туманная» платформа развертывается близко к конечным пользователям – между традиционными giant-like облачными ЦОДами и конечными пользователями. Хотя как облачная, так и «туманная» парадигма поддерживает почти аналогичный набор сервисов (вычисление, хранение, сетевое взаимодействие), между ними существуют разли-

чия. Развертывание «туманных» вычислений предназначено только для определенного географического региона. Кроме того, эта платформа специально создавалась для приложений IoT и для приложений, требующих ответа в реальном времени с минимальной задержкой. С другой стороны, традиционный облачный ЦОД централизован и расположен в основном далеко от пользователя. Ему присущи некоторые ограничения по задержке и времени отклика приложений реального времени. «Туманная» платформа предполагает использование разнообразных устройств, собирающих данные разных типов. Взаимодействие между гетерогенными устройствами – не единственная проблема для «туманных» вычислений. Другими являются оркестрация и управление ресурсами, их балансировка, масштабируемость, безопасность и конфиденциальность. До сих пор нет стандартной модели монетизации для «туманных» вычислений. Это все еще открытая исследовательская проблема.

Cloudlets (облачка) разрабатываются командой Университета Карнеги – Мелона [11], [28]. Они предназначались для приближения облачных сервисов к мобильным пользователям. Внутри «облачка» состоят из набора достаточно мощных ресурсов, таких как многоядерные серверы с высокоскоростным подключением к Интернету и высокоскоростной беспроводной локальной сетью для связи с мобильными устройствами [11]. Мобильное устройство, рассматриваемое как тонкий клиент, может загружать вычислительные задачи через беспроводную сеть в «облачко», находясь при этом от него на расстоянии одного скачка (hop). Наличие облачка вблизи мобильного устройства необходимо для сокращения и предсказуемости времени приема-передачи для исполняемых приложений. Если устройство выходит из зоны действия «облачка», то оно либо должно переключиться на удаленный облачный ЦОД или пользоваться своими собственными ресурсами.

Концепция Micro-DC (μDC; микро-ЦОД) была представлена компанией Microsoft Research [10]. Она рассматривает микро-ЦОД как расширение сегодняшнего большого облачного ЦОД. Подобно «облачкам», микро-ЦОД также разработан для приложений, которые требуют малых задержек на коммуникацию, обработку, и устройств, которые работают в условиях жестких ограничений по энергетике. Микро-ЦОД – это вычислительный комплекс, состоящий из одного или нескольких соединенных между собой стоек, оснащенных всей необходимой инфраструктурой для ИТ-оборудования, собранных и протестированных производителем. Микро-ЦОД представляет собой автономную безопасную вычислительную среду, которая включает в себя все необходимые вычислительные ресурсы, хранилище данных и сетевое оборудование для работы клиентских приложений. Потребляемая мощность микро-ЦОД может составлять от 1 до 100 кВт для удовлетворения требований к масштабируемости и задержкам с учетом рабочей нагрузки, она может меняться, если в будущем потребуется больше энергии.

Следует отметить, что промышленность давно освоила производство таких вычислительных комплексов. Примерами являются UCS от Cisco, V-Blocks компании VCE, Active Systems от Dell, компания Schneider Electric предлагает Smart Bunker и Smart Data Safe. Система VPLEX является продуктом компаний EMC, Microsoft и AVNET [28]. Huawei – еще один производитель, выпускающий микро-ЦОДы [20]. Перечисленные выше микро-ЦОДы в основном предназначены для размещения вычислительных ресурсов и телекоммуникационного оборудования в неподготовленных помещениях (таких как офисы, склады, подсобные помещения или производ-

ственные объекты) и подключения их в корпоративную сеть. По словам поставщиков, время их установки (до начала использования) сократилось до 60–70% по сравнению с классическим решением.

Mobile Edge Computing (MEC; мобильные периферийные вычисления) предназначены для предоставления облачных вычислительных ресурсов и ИТ-услуг на периферии сотовых сетей [14] (см. Рис. 1). MEC обеспечивает малые задержки, близость к оконечному устройству пользователя, знание контекста его работы и местоположения, а также более высокую пропускную способность. Как видно на рисунке 1, серверы MEC развернуты на сотовых базовых станциях, что позволяет гибко и быстро развертывать новые приложения и услуги конечным пользователям. MEC можно рассматривать как облачные серверы, работающие на границе зоны радиодоступа мобильных сетей и реализующие конкретные услуги, которые не могут быть достигнуты с использованием традиционной сетевой инфраструктуры. При использовании MEC весь трафик перенаправляют не на удаленный облачный ЦОД, а на серверы MEC. Таким образом, серверы MEC, работающие с приложениями и выполняющие связанные с ними задачи обработки данных, ближе к сотовым клиентам, уменьшают нагрузку на сети и время отклика приложения. ETSI1 разработал отраслевую спецификацию MEC [21] и опубликовал ее в сентябре 2014 года. Были предложены системная архитектура и стандарты ряда API, необходимые для MEC [21]. Компания Nokia, например, продемонстрировала, что MEC играет ключевую роль в автоматизации вождения автомобилей. В случае подключения автомобиля к традиционному облачному ЦОД задержки будут составлять не менее 100 мс. Базовые станции с распределенными облаками MEC продемонстрировали сквозную задержку в пределах 20 мс.

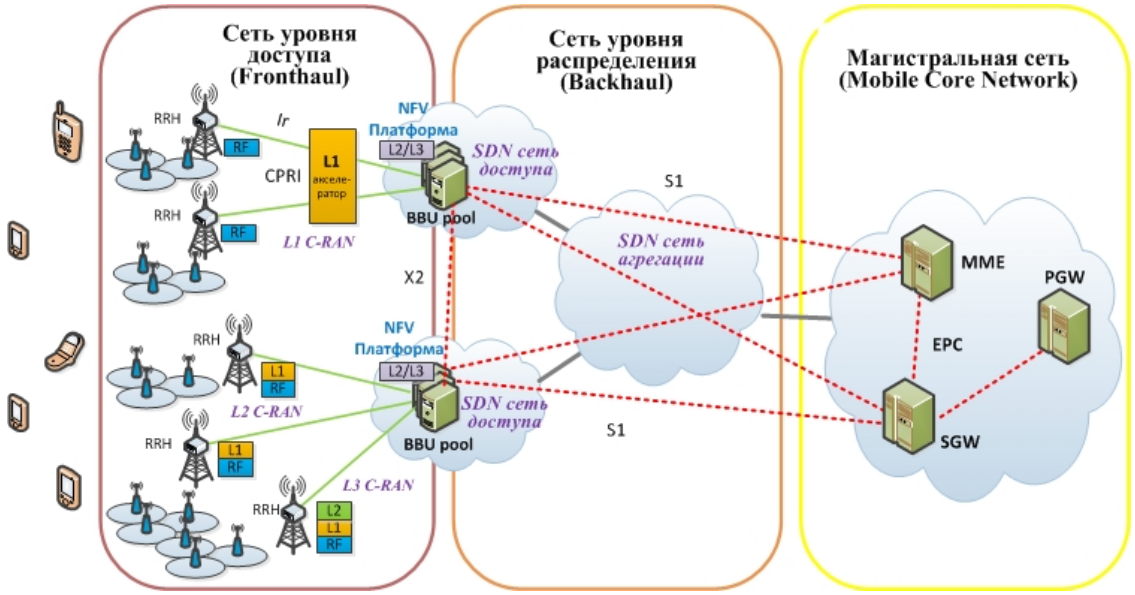

Рис. 1. Mobile Edge Computing для сетей 5G

Fig. 1. Mobile Edge Computing for 5G networks

Анализ приложения CDN. Технологии периферийных вычислений, описанные выше, подходят для тех приложений, для которых можно локализовать необходимые сервисы в непосредственной близости от мобильного устройства, чтобы удовлетворить ограничениям, связанным с задержками или с недостатком вычислительной мощности. Однако есть приложения, для которых ни один из вышеупомянутых вариантов организации ИКТ-инфраструктуры не является достаточным. Например, согласно данным компании Akamai [30], более 650 сетей участвуют в доставке 90% трафика этой компании. Если мы посмотрим на приложение доставки контента как на пример приложения реального времени, то мы увидим, что в доставке этой услуги участвуют несколько разных сетей разных провайдеров.

Рассмотрим в качестве примера организацию трансляции игр на чемпионате мира по футболу 2014 года в Бразилии [23]. Для этого чемпионата была создана специальная инфраструктура по производству видеоконтента, включающая следующие услуги, поддерживаемые компанией EVS (Event Video Service):

- Мультимедийная прямая трансляция матча;
- Выделенный мобильный/мультимедийный канал;
- Дополнительный контент в режиме видео по запросу (Video-on-Demand; VoD);
- Многоракурсный контент;
- Вставка рекламных видеоклипов в контент;
- Мультимедиа и текстовые сообщения (MMS и SMS);
- Интерактивный доступ к данным;
- Визуализация данных.

Техническая инфраструктура EVS объединила множество ведущих технологических решений нескольких компаний, которые работали вместе, предлагая для зрителей лучшие интерактивные и мультимедийные продукты:

- от компании Elemental Cloud для облачной обработки видеопотока в реальном времени;
- от компании Aspera для передачи файлов на высокой скорости через сети от места проведения мероприятия до облачной инфраструктуры;
- от компании Brightcove – облачные сервисы для операций по перекодированию мультимедийного контента;
- от Amazon S3 – хранилище данных;
- от Akamai – сеть доставки контента (CDN);
- от компании NETCO Sports - сервис режима второго экрана («second screen»).

Во время проведения матчей шесть видеопотоков с шести камер (camera angles) записывались на серверах EVS XT3 непосредственно на месте проведения матча, где автоматически обрабатывались средствами EVS C-Cast Agent и отправлялись в виде потоков по 10 Мбит/с каждый компанией IBC (Международная вещательная корпорация) через волоконно-оптическую сеть. В компании IBC они обрабатывались специальным программным обеспечением C-Cast Central, после чего обработанные потоки передавались по волоконно-оптической сети из IBC в хранилище Amazon S3 в Дублине, где было развернуто производство EVS C-Cast. Там каждый входящий поток, при помощи программного обеспечения компании Elemental, был фрагментирован на пакеты, каждый из которых нёс видеофрагмент с фиксированным временем проигрывания. После этого из каждого такого пакета генерировали 10 различных видеопотоков разного качества, которые передавали со скоростью 10 Мбит/с для доставки по CDN сети по конкретному адресу.

В рассматриваемом примере отметим следующее. Сеть доставки контента (CDN) имеет статическую оверлейную структуру, производство 10 видеопотоков различного качества для передачи одного и того же контента было распределено между конкретными устройствами, кэширование производилось только на Edge серверах. Если бы такие сервисы для видеопотоков, как перекодирование, кэширование, трансрейтинг, компрессия, были доступны не только на Original и Edge серверах, но и в сетях, через которые велась трансляция, это значительно снизило бы нагрузку на эти серверы. Также за счет интенсивного использования групповой передачи (multicast) вместо одноадресной (unicast) передачи сократилась бы и нагрузка на сеть. Следует отметить, что эффект от размещения всех перечисленных сервисов в сети значительно возрос бы, если бы размещение определенных видов сервисов в сети можно было менять динамически. Примером такого сервиса может служить кэширование. В зависимости от степени популярности, контент должен быть кэширован как можно ближе к какой-либо локальной группе клиентов, если этот контент представляет интерес в значительной степени для этой локальной группы клиентов. Если же он будет интересен более широкой аудитории, то размещать его надо так, чтобы время доступа и объем передаваемого трафика для клиентов из разных регионов были сбалансированными.

Из рассмотренного примера видно, что между сетями, в которых размещается источник данных и/или инициатор запроса данных, было задействовано несколько сетей с дополнительными услугами по обработке видеопотоков. Возникает вопрос: где та «граница», периферия, о которой мы говорим? Границу какой сети мы имеем в виду, когда говорим о доставке контента?

Другой важный вывод, который можно сделать на основе рассмотренного – приложение более не локализовано в одном ЦОДе. Оно превратилось в систему взаимодействующих сервисов, распределенных, в общем случае, в разных сетях.

2. Концепция иерархических периферийных вычислений

На рисунке 2 показана типичная структура сети доставки контента (CDN), арендованной поставщиками контента (CP) у какого-либо Интернет-провайдера.

Как видно из этого рисунка, существует несколько сетей, подконтрольных различным Интернет-провайдерам, которые предоставляют клиентам доступ к контенту определенного поставщика контента. Естественно, возникает вопрос, о какой периферии какой сети идет речь, какие из этих Интернет-провайдеров и как должны взаимодействовать при доставке контента?

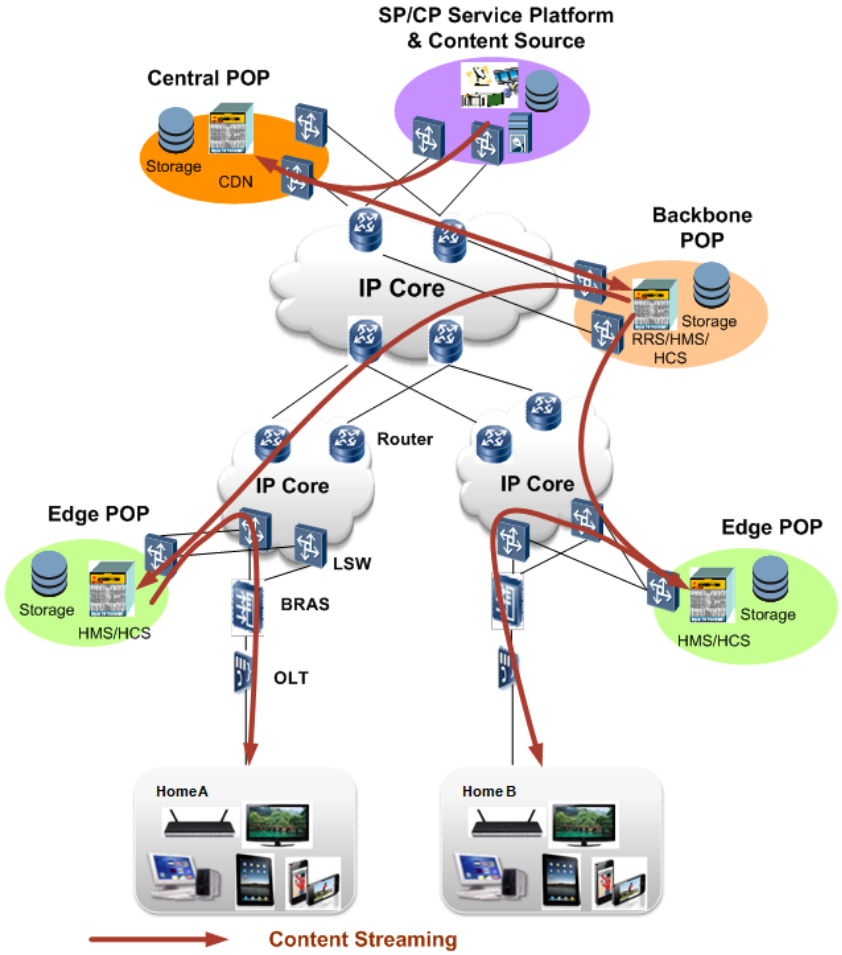

Рис. 2. Типовая структура CDN сети,
арендуемая контент-провайдером (CP) у ISP оператора

Fig. 2. Typical structure of a CDN network
rented by a content provider (CP) from an ISP operator

Из этой иллюстрации и приведенного выше примера можно сделать вывод:

- Во-первых, данные для приложения могут поступать из нескольких источников, расположенных в разных сетях;
- Во-вторых, ограничение на время доставки, обработки полученных данных не всегда может быть соблюдено в случае единого центра обработки данных. В этом случае решением может быть распределение обрабатывающих

мощностей в нескольких периферийных центрах обработки, в сетях разных Интеренет-провайдеров (ISP операторов) с учетом ограничений на время их взаимодействия.

- В-третьих, нужны динамизм и гибкость при выборе топологии виртуальной CDN сети и таких ее параметрах, как пропускная способность, места размещения в ней таких сервисов, как кэширование, сжатие, транскодинг, трансрейтинг и т.д.

Идея иерархических периферийных вычислений основана на размещении микро- или мини-ЦОД (μ/m-DC соответственно), оснащенных сетевыми, вычислительными ресурсами, ресурсами хранения данных, на периферии сетей, через которые проходит поток данных от/до устройств конечного пользователя. Такие сети мы будем называть взаимодействующими сетями. Устройства конечных пользователей обычно представляют собой смартфоны, носимые гаджеты и различные устройства для «интернета вещей», беспроводных сенсорных сетей, требующие реакции в реальном времени. Концепция иерархических периферийных вычислений позволяет динамически выбирать распределение затрат на передачу и обработку данных в зависимости от текущей загрузки ресурсов взаимодействующих сетей, расположенных между источником данных/запроса и местами их обработки, и ограничений, накладываемых приложением.

Развертывание как вычислительных ресурсов, так и ресурсов хранения данных на границах взаимодействующих сетей позволяет реализовать большое количество приложений, требующих реакции в реальном времени. Список таких приложений включает в себя, но не ограничивается перечисленными ниже примерами:

- мониторинг транспортных потоков и навигация, включая передачу информации о трафике и расчет маршрутов для определенного региона непосредственно на периферии сети;

- фильтрация и агрегация данных, при которых выполняется предварительная фильтрация контента и данных на периферии сети, для уменьшения объема данных перед отправкой их в традиционное облако;

- дополненная реальность, интерактивные медиа, распознавание речи, обработка естественного языка [11], [28];

- сеть доставки контента (CDN);

- научные эксперименты.

Важным свойством концепции иерархических периферийных вычислений является возможность динамически определять состав и расположение сервисов в виртуальной ИКТ-инфраструктуре с использованием техники виртуализации сервисов в облаках микро- и мини-ЦОДов на периферии взаимодействующих сетей. Под термином «периферия взаимодействующих сетей» мы будем понимать набор точек входа/выхода в сеть конкретного провайдера, например, точку присутствия провайдера (PoP) в зоне доступа его сети, BGP шлюзы и т.д. Как можно ближе к

этим точкам необходимо разместить микро- и мини-ЦОДы с облачной инфраструктурой, с вычислительными мощностями и хранилищами данных. Термин «ближе» означает контролируемую, небольшую и гарантированную задержку доступа. Здесь мы также будем широко трактовать понятие услуги. В сети все является сервисом: предоставление ресурса – сервис, агрегирование функций обработки – сервис. Такую трактовку сегодня позволяет подход концепций программно-конфигурируемой сети (ПКС) и виртуализации сетевых функций (ВСС).

Для иерархических периферийных вычислений ПКС предлагает уникальную возможность поддерживать глобальное видение сети вплоть до соединений и портов, различных полей заголовков разного уровня. При этом контроллеры ПКС могут запрашивать статистику коммутаторов, обнаруживать потоки с определенными шаблонами заголовков пакетов, динамически реагировать на обнаруживаемые угрозы. Виртуализация сетевых функций предоставляет уникальные возможности динамически размещать виртуализованный сервис в надлежащем периферийном ЦОДе, чтобы поддерживать баланс между задержкой и временем обработки. Обе концепции хорошо дополняют друг друга.

Одной из ключевых проблем такой сетевой организации становится взаимодействие автономных систем (АС) для обмена информацией о том, какие услуги они могут предоставить друг другу, с какими характеристиками, с каким качеством. Традиционно АС обмениваются информацией в точках обмена Интернет-трафиком (IXP), которая в случае использования ПКС называется Software Defined Exchange Point (SDX).

Сегодня основным протоколом, обеспечивающим связь между автономными системами, является BGP. Однако для иерархических периферийных вычислений BGP протокол в традиционной форме не подходит. В своей традиционной форме он предназначен для выбора маршрута для доступа из одной АС к другой АС. Этого для реализации концепции иерархических периферийных вычислений уже недостаточно. Необходимо, чтобы протокол, назовем его «расширенным BGP» (Extended BGP – EBGP), разрешал АС (можно в этом случае использовать термин АС-издатель (AS-publisher)) сообщать другим, внешним АС, какие услуги могут быть им доступны в АС-издателе и в соответствии с какими соглашениями об уровне услуг (SLA). АС-издатель должен предоставлять о каждой опубликованной услуге такую информацию, как: политика выставления счетов, соглашение об уровне услуг, информация о качестве обслуживания (QoS), минимальная ожидаемая производительность (задержка на обработку), через какой шлюз АС-издателя эта услуга доступна. Здесь термин «услуга» можно толковать широко: это предложение некоторых ресурсов, инфраструктуры как услуги (IaaS) и различных виртуализированных сервисов. Ключевым моментом является то, что АС-издатель должен гарантировать условия соглашения об уровне услуг, такие как: доступность услуги, задержка доступа, джиттер, производительность. Одним из возможных способов сделать это является включение в объявление АС некоторой части спецификации TOSCA (Topology and Orchestration Specification for Cloud Applications).

Язык спецификации TOSCA предназначен для предоставления информации о виртуализированной услуге, представляющей собой композицию из виртуализированных функций [33] в виде шаблона услуги. Шаблон описывает различные типы политик услуги, такие как масштабирование, пороговые величины (критическая

загрузка процессора, топология, производительность), некоторые параметры виртуальной машины (число ядер, оперативная память, объем диска, операционная система и т.д.). В описании политики также задается группа узлов сети, для которых будут отслеживаться эти параметры. Это описание также содержит инструкции, что делать, когда критическое значение будет достигнуто хотя бы на одном из узлов из группы. Подробнее см. [29].

Механизм шаблонов TOSCA – это средство описания, спецификации сервиса. Для целей объявления сервиса и связанной с ним политики можно было бы применить средства, созданные в рамках концепции сервис-ориентированной архитектуры [32], такие как язык WSDL (Web Services Description Language), для описания интерфейсов службы, протокол SOAP (Simple Object Access Protocol) для описания формата получаемых и отправляемых сообщений, а также стандарт UDDI (Universal Description, Discovery and Integration) для создания каталогов доступных услуг. Таким образом, концепция иерархических периферийных вычислений позволяет динамически выбирать затраты на передачу и обработку данных в соответствии с текущим уровнем загрузки ресурсов сетей, расположенных между источниками данных и/или запросов данных и местами их обработки, с учетом ограничений, налагаемых приложениями.

Важным моментом любой технологии является ее «уживчивость» с уже существующими. Можно привести много примеров, когда технология с прекрасными характеристиками так и не находила себе практического применения, просто потому, что ставила вопрос: либо я, либо все остальные. Отсутствие стандартов и устоявшейся терминологии приводит к неправильным представлениям о соотношениях между технологиями периферийных вычислений, «интернетом вещей» и облачными вычислениями. Примеры таких заблуждений упоминаются в литературе, где авторы утверждают, что технологии периферийных вычислений будут «теснить» или «заменять» облако туманом или децентрализованной парадигмой облачных вычислений на периферии. Как упоминалось в [28], существует путаница в понимании передовых Edge технологий, например, некоторые авторы рассматривают туманные вычисления как микро-ЦОДы [12], [13], в то время как другие сосредоточены главным образом на идее усиления и оснащения сетевых компонентов дополнительными ресурсами обработки и хранения.

Нужно четко понимать, что технологии периферийных вычислений, включая иерархические периферийные вычисления, не следует рассматривать как замену облачной парадигме. Как показано на рисунке 3, эти технологии дополнят облако и расширят облачные сервисы до самых отдаленных закоулков сетей, так чтобы удовлетворялись потребности приложений по работе в реальном времени. Таким образом, подход иерархических периферийных вычислений не конкурирует ни с «туманными» вычислениями, ни с «облачками», ни с мобильными «граничными» вычислениями, или микро-ЦОДами, ни с традиционным подходом, основанным на giant-like ЦОДах. Напротив, они дополняют друг друга. Это ясно видно на рисунке 3, где представлен список некоторых потенциальных областей применения иерархических периферийных вычислений.

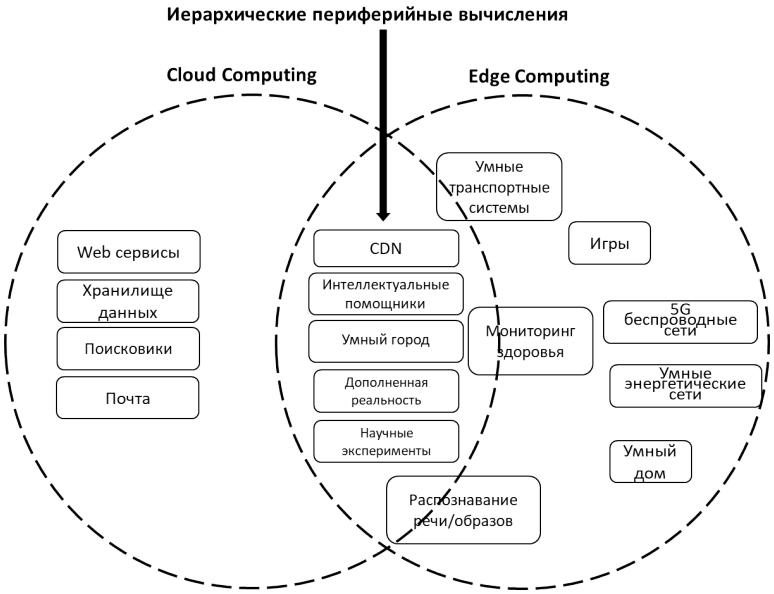

Рис. 3. Соотношение приложений традиционного подхода к облачным вычислениям на основе giant-like ЦОД, технологий Edge Computing и концепции иерархии периферийных вычислений

Fig. 3. Relation among applications of the traditional approach to cloud computing based on giant-like data center, Edge Computing technologies, and the concept of Hierarchical Peripheral Computing

3. Потенциал иерархических периферийных вычислений

Организация ИКТ-инфраструктуры на основе концепции иерархических периферийных вычислений позволяет:

1. Управлять задержкой при взаимодействии пользователя с сетевыми приложениями.

2. Сократить трафик через взаимодействующие сети.

3. Сократить требования к вычислительным возможностям и объему памяти на конечных устройствах (терминалах, мобильных устройствах и датчиках).

4. Минимизировать потребление энергии как на мобильных устройствах, так и на датчиках/сенсорах «интернета вещей».

Например, минимизация трафика в магистральной части одной из промежуточных сетей достигается путем помещения услуг по обработке входящего трафика как можно ближе к точке его входа. Очевидным примером тому является пример сети доставки контента (CDN). В случае «интернета вещей» минимизация трафика в опорной сети достигается за счет предварительной обработки трафика от датчиков на границе сети, т.е. ближе к датчику.

Организация иерархических периферийных вычислений на основе микро- и/или мини-ЦОД позволяет:

1. Свести к минимуму капитальные затраты, используемые площади и энергопотребление такого ЦОД.

2. Значительно сократить время на создание микро-ЦОД и ввод его в эксплуатацию.

3. Уменьшить сроки возврата инвестиций при создании микро-ЦОД.

Сокращение трафика в сети. Традиционная модель взаимодействия пользователя с Интернетом предполагает короткие запросы от пользователей на доступ к Интернет-сервису и получение в ответ иногда больших объемов данных. Например, передача файлов и, в частности, видео по запросу (VoD) или передача потокового видео в реальном времени, состоят из очень маленьких запросов данных от пользователя к поставщику услуг (SP) и большого объема данных, поступающих от поставщика услуг к пользователям. Как мы уже говорили, кэширование видеоконтента на границе разных сетей, образующих инфраструктуру для сети доставки контента (CDN), может значительно сократить объем передаваемых данных и задержки передачи контента от поставщика услуг до пользователя.

Динамическое размещение кэшей в сети доставки контента также может уменьшить эту задержку. Например, наиболее популярный контент сохраняется в кэшах Интернет-провайдеров или в сетях доставки контента (CDN), чтобы свести к минимуму поток данных и задержки во взаимодействующих сетях [24] – [26]. Транскодинг видео на периферии CDN в максимальной близости к конечным пользователям/устройствам позволяет использовать групповую рассылку вместо одноадресной передачи, что значительно сокращает объем трафика во взаимодействующих сетях. Это также помогает минимизировать задержки доступа и максимизировать «качество восприятия» (QoE) зрителей.

Например, при потоковом вещании, в частности потоковом вещании с использованием краудсорсинга, значительные объемы данных передаются от пользователей сервис-провайдеру, а затем распространяются глобально через различных провайдеров, таких как Twitch (on-line игровая система с поддержкой краудсорсинга), YouTube Live [28], Periscope [28] и YouNow [28]. Компания Netflix размещает свою огромную коллекцию развлекательного видеоконтента. Если 10% из 8 миллионов человек в Нью-Йорке захотят одновременно просматривать фильмы от Netflix, для одновременной обработки всех запросов потребуется инфраструктура с пропускной способностью 1,6 Терабит в секунду (Tbps) [28].

Например, рассмотрим, как вариант, финальный матч европейского футбольного турнира, где сеть Akamai обслуживала 3,3 миллиона видеопотоков одновременно, испытывая пиковую нагрузку в 7,3 Тбит/с [22]. В условиях отсутствия групповой рассылки, что является сегодня типичным случаем для CDN сетей из-за проблем с конфигурацией и безопасностью, такой поток данных, проходящий через множество маршрутизаторов между сетью доставки контента и сетью провайдера, приводит к значительным затратам энергии, а также к затратам на поддержание такой сети и управление ею. Кроме того, сеть доставки контента имеет пассивную систему хранения, хранящую большие объемы данных, и очень ограниченные возможности для

обработки данных. Транскодинг видеороликов на лету недоступен в существующих сетях доставки контента. В то же время, кэширование на границе сотовой сети (например, на базовой станции или на eNodeB) может сэкономить значительное количество трафика в Backhaul сегменте сети.

Минимизация задержки. Как отмечено в [28], для сервисов визуальной навигации (visual guiding services) в режиме реального времени предпочтительное время отклика составляет от 25 мс до 50 мс. Задержки в облачных вычислениях в Интернете являются серьезной проблемой для приложений, которым требуется реакция в реальном времени, например, таким как интеллектуальные транспортные системы, игры, приложения, использующие потоковое видео, и другие, важные для безопасности приложения, где такие задержки недопустимы. При использовании традиционного облака задержка на передачу данных от клиента к поставщику облачных услуг может составить от нескольких миллисекунд до секунды (см. Таблицу 1 выше) [28]. Даже небольшая задержка при обработке запроса пользователя может привести к потере абонента и дохода. Например, в [15] отмечено, что замедление обработки запросов всего на 2 секунды привело к сокращению количества запросов на одного пользователя на 1,8% и дохода на пользователя на 4,3%. Опрос, проведенный Forester, показал, что большинство покупателей Интернет-магазинов считают время отклика веб-сайта основным фактором при оценке удовлетворенности [28]. В этом опросе также выяснилось, что более 40% клиентов могут ждать загрузку страницы не более трех секунд, прежде чем они покинут сайт. В обзоре IDC [16] сообщается, что улучшение производительности и надежности услуг предоставляемых компанией Akamai и ориентированных на ускорение корпоративных приложений привело к ежегодному увеличению доходов этой компании с 2 до 3 миллионов долларов США. Поэтому размещение контента у локальных Интернет-провайдеров (граница сети) имеет решающее значение для областей с низкой связностью и высоким временем отклика [30]. В настоящее время многие Интернет-провайдеры создают облачные инфраструктуры на границах своих сетей. В таблице 1 выше показано, что иерархические периферийные вычисления могут снизить задержки, уменьшить трафик в опорной части сети, если виртуализированные сервисы будут расположены на границе сети Интернет-провайдера.

Сокращение потребления энергии. Потребление энергии облачными сервисами обычно зависит от следующих факторов [28]: (а) потребление энергии устройством конечного пользователя, (б) потребление энергии в ЦОДах, используемых для иерархических периферийных вычислений, включая энергию, потребляемую во внутренней сети самого ЦОДа, а также хранилищем данных, (в) объем трафика, передаваемого между пользователем и облаком, использующим иерархические периферийные вычисления, (г) вычислительная сложность выполняемой задачи; (д) количество пользователей, использующих вычислительный ресурс, и (е) потребление энергии транспортной сетью.

В [18] показано, что 14% потребления энергии в Интернете связано с транспортировкой данных. Этот источник также показывает, что интерактивные приложения генерируют значительный объем трафика и потребляют больше энергии из-за накладных расходов, возникающих в результате взаимодействия в реальном времени с традиционным облачным ЦОД. Было показано, что большой дополнительный трафик связан с частой установкой/разрывом TCP сессий и объемом данных, передава-

емых при этом пользователю и от пользователя за сеанс (измеряется от десятков до сотен КБ) [28]. Как мы уже говорили, кэширование на краю сети позволяет значительно сократить задержки доступа и сетевой трафик [28]. Кроме того, технологии периферийных вычислений позволяют размещать специализированные сервисы на границе сети для обеспечения реакции в реальном времени и фильтрации данных. Например, компания Akamai развернула пограничные вычислительные сети для обеспечения распределенного выполнения Java-приложений [30].

Сектор ИКТ является одним из основных потребителей электроэнергии, который, по оценкам, потребил более 271 млрд. КВт/ч энергии в ЦОДах в 2010 году [3]. Сетевая инфраструктура также является одним из основных потребителей электроэнергии. По оценкам [2], в 2010 году она потребила около 15,6 млрд. КВт/ч энергии. Эксперименты с периферийными вычислениями на границах взаимодействующих сетей, через которые проходит трафик мобильных устройств, показали, что потребление энергии может быть уменьшено на 42% [27].

Вычисления, организованные по принципу иерархических периферийных вычислений, могут дополнить традиционные облачные вычисления для определенных приложений, и это может привести к экономии энергии, если приложение или его компоненты могут быть перенесены из традиционного ЦОД в микро-ЦОД в транзитной сети. Кроме того, кэширование данных на граничных устройствах снижает нагрузку на опорную сеть, что позволяет снизить пропускную способность каналов за счет таких «зеленых» технологий, как Adaptive Link Rate (ALR) [4].

Уменьшение нагрузки на традиционные облака и традиционные ЦОДы. Сервисы, с определением своего местоположения (Location aware services), ежедневно генерируют огромное количество данных. Популярны несколько приложений для регистрации и учета спортивной активности, таких как Nike +, Runtastic, Runkeeper и Endomondo [28]. Эти приложения работают на смартфонах и регистрируют ежедневную активность пользователей с помощью различных датчиков, например, акселерометров, GPS, гироскопов и датчиков температуры, обычно устанавливаемых на смартфонах. В основном данные, записанные приложениями, отправляются в облако в виде наборов, где каждый набор содержит несколько записей, таких как идентификатор пользователя, долгота, широта, время, расстояние, скорость, продолжительность, калории, погода и другие параметры. Например, недавнее исследование Endomondo показало, что за одну часовую тренировку в среднем генерируется 170 GPS-записей, а среднее количество записей, генерируемых в месяц, составляет от 2,8 до 6,3 млрд. [17]. С 30 миллионами пользователей число записей, генерируемых в секунду, может достигать 25 000 [17]. Этот огромный объем данных будет закачиваться в облако умным городом. Более того, не все полученные данные полезны. Например, датчики, развернутые в проекте Большого адронного коллайдера (LHC), генерируют около 500 Экзабайт данных в день. Однако 99,999% этих данных отфильтровываются [17]. Поставщики приложений, чтобы отфильтровать локально ненужные данные и обеспечить реакцию в реальном времени для пользователей, находящихся поблизости от точки доступа в сеть, могут использовать периферийные вычисления. Более того, т.к. данные будут отфильтрованы до отправки в облако, то сетевой трафик и нагрузка на обрабатывающие эти данные облачные серверы будет сокращена.

Другим хорошим примером, где технология иерархических периферийных вычислений позволяет успешно справляться с большими объемами трафика и временными ограничениями, являются такие потоковые приложения, как Facebook Live, YouTube Live и Livestream [28], позволяющие пользователям осуществлять прямую трансляцию. Сообщается [6] , [31], что в течение одной минуты пользователи YouTube загружают 72 часа нового видео, пользователи Facebook выкладывают в сеть 2 460 000 фрагментов контента, пользователи WhatsApp выкладывают 347 222 фотографии, пользователи Instagram публикуют 216 000 новых фотографий, а пользователи Vine – 8333 видеороликов. Обычно, когда видео или фото загружается, например, на Facebook или YouTube, для уменьшения размера изображения оно подвергается сжатию с потерями. Загрузка фотографий и видео с высоким разрешением непосредственно с пользовательских устройств в облако потребует значительную долю полосы пропускания канала и может занять много времени в тех областях, где используется Интернет-соединение плохого качества.

Подобные проблемы возникают в приложениях мониторинга состояния здоровья или в приложениях интеллектуального управления городом, где потоки данных с камер наблюдения и других датчиков необходимо загружать в облако. Технология иерархических периферийных вычислений может быть использована для переноса задач, связанных со сжатием данных перед загрузкой в облако, на периферийные мини- или микро-ЦОДы поблизости от конечных пользователей. Более того, в этих ЦОДах также может выполняться шифрование пользовательских данных вместо загрузки необработанных данных в облако, что обеспечивает безопасность и конфиденциальность пользовательских данных при транзитной передаче.

Итак, построение решений для «интернета вещей» в двухуровневой архитектуре с традиционным giant-like облачным ЦОДом на одном конце и устройствами для «интернета вещей» на другом не удовлетворяет требованиям по величине задержек, скорости перемещения терминала (мобильности) и точности определения местоположения терминала [19]. Как отмечалось ранее, удовлетворить перечисленные выше требования позволяет многоуровневая архитектура иерархических периферийных вычислений, показанная на рис. 4.

На первом уровне этой архитектуры находятся сенсоры, датчики сбора данных для приложений «интернета вещей», либо сами приложения, развернутые на соответствующих устройствах, которые являются устройствами конечного пользователя, например транспортным средством.

Вторая часть архитектуры – граничный ТУМАН, связанный с датчиками, конечными пользователями через маршрутизатор, точку доступа, сеть беспроводного доступа или базовую станцию LTE.

Третья часть архитектуры иерархических периферийных вычислений представляет собой сеть неоднородных ЦОДов: облака на периферии взаимодействующих сетей на основе сетей микро-ЦОДов и, наконец, традиционные большие облачные ЦОД.

Организация доступа на границе беспроводных сетей на основе технологий ПКС и ВСС (SDN NFV), равно как и построение опорной (core) части сотовых сетей 5G на основе виртуальных сетевых функций, пока также требует проработки и исследования.

Рис. 4. Архитектура иерархических периферийных вычислений

Fig. 4. Architecture of Hierarchical Peripheral Computing

Заключение

Сегодня технология иерархических периферийных вычислений только зарождается. Нет общепринятой архитектуры, устоявшейся системы понятий, стандартов и протоколов. На рисунке 5 показаны типичные приложения, для которых необходима архитектура иерархических периферийных вычислений. Там же перечислены потенциальные возможности этой архитектуры, которые были обсуждены выше. Благодаря им, эффективность наших сетей может быть значительно повышена. Однако для того, чтобы сделать этот подход осуществимым, необходимо решить много проблем, провести исследования. Некоторые из этих проблем перечислены здесь. Этот список не полон, те, что перечислены, не охватывают их всех.

Безопасность и приватность. Говоря о безопасности иерархических периферийных вычислений, мы должны понимать, что речь идет о безопасности в неоднородной облачной среде, безопасности в контуре управления и в контуре передачи данных ПКС сетей, безопасности, связанной с использованием предложенными выше расширениями протокола BGP – EBGP, уязвимостях, которые могут привнести виртуализованные сервисы от независимых поставщиков [36]. В некоторой степени проблемы безопасности в контуре управления ПКС сравнительно проще из-за централизованного характера и логически единого «органа» управления сетью – контроллера. Важными проблемами являются безопасность как самого ПКС контроллера, так и его приложений, изоляция приложений от взаимного влияния как в контуре управления, так и в контуре передачи данных, выявление скомпрометированных коммутаторов в контуре передачи данных ПКС сетей. Скомпрометированные коммутаторы являются важной угрозой, поскольку они могут быть использованы для разнообразных сетевых атак.

Рис. 5. Приложение, возможности и проблемы иерархических периферийных вычислений

Fig. 5. Application, possibilities and problems of Hierarchical Peripheral Computing

Одним из механизмов, который может быть использован для обнаружения скомпрометированных коммутаторов в ПКС сетях, является прогнозирование значения счетчиков правил коммутации пакетов в коммутаторе. Поскольку контроллер обладает полной информацией об использовании правил коммутации и обработки пакетов в коммутаторе, равно как и о самих правилах, определенных в сети, то можно обнаружить аномальное поведение в сети, предсказав ожидаемые значения счетчиков и сравнив их с реальными значениями счетчиков, полученными контроллером от коммутаторов [34, 35]. На периферии сети миграция сервисов из одной сети в другую, развернутых разными операторами сервисов, может приводить к возникновению уязвимостей, которые пока слабо изучены.

Необходимо также тщательно изучить проблемы обеспечения приватности (privacy), связанные с передачей данных от пользователя в сеть и передачей их из периферии одной сети на периферию другой. Дело в том, что устройства пользователей могут быть недостаточно мощными для надежного шифрования. В таком случае эту задачу могу решить устройства на периферии ближайшей сети. Однако соединение между устройством конечного пользователя и микро-/мини-ЦОДом на периферии ближайшей сети становится удобным местом для организации «атаки чужой в середине». Кроме того, шифрование увеличивает задержку на передачу данных до места их обработки. Однако здесь важнее то, что большинство конечных пользователей обычно не задумываются о приватности и безопасности своих данных. Необходимы автоматические механизмы, обеспечивающие приватность пользовательских данных во время работы. Например, в опросе [6] выяснилось, что 80% из 439 миллионов пользователей Wi-Fi сетями используют в своих беспроводных маршрутизаторах пароли, установленные по умолчанию, а 49% пользовательских

сетей не защищены. Кроме того, 89% общедоступных Wi-Fi точек доступа являются незащищенными. В [9] сообщается, что к 2020 году 10% всех атак будут нацелены на системы «интернета вещей». Также важно сохранять приватность данных пользователя и изолировать их от других данных, собранных сторонними приложениями. Так, данные от приложения, отслеживающего физическую активность пользователя, не должны смешиваться, а само приложение иметь доступ к данным о других видах деятельности пользователя или, например, данным о состоянии электроприборов в его умном доме [7]. Поддержка механизмов контроля доступа к данным разных пользователей на периферии разных сетей – это еще одна проблема обеспечения приватности данных.

Стандартизация архитектуры и протоколов. ПКС – довольно новая концепция сетевой архитектуры. Для ее широкого применения требуется стандартизация. Эти стандарты необходимы для проектирования, интеграции, эксплуатации и технического обслуживания таких сетей. Необходимо разработать политику для взаимодействия доменов ПКС, находящихся под управлением разных контроллеров, друг с другом и с другими, например, унаследованными (legacy) доменами традиционных сетей, а также с виртуализованными сервисами в ПКС сети ЦОДа (там может быть свой контроллер). В контуре управления ПКС сетью используются три интерфейса или API: 1) южный интерфейс, 2) северный интерфейс и 3) интерфейс восток-запад. Интерфейс восток-запад определяет, как контроллеры ПКС доменов взаимодействуют друг с другом для обмена информацией в процессе управления. По сути, мы можем думать об этом интерфейсе как о канале, проходящем через разные ПКС домены, для связи с их контурами управления. Организация интерфейса восток-запад оказывает прямое влияние на отказоустойчивость системы управления ПКС и доступность сервисов, реализуемых приложениями ПКС контроллера. Здесь широкое поле для исследований [36]. В целях масштабируемости, доступности и надежности сервисов в контуре управления сетью необходимо обеспечить консистентность данных в физически распределенном контуре управления, но который является логически централизованным. Это необходимо для того, чтобы, прежде всего, гарантировать, что резервный контроллер сможет корректно принять на себя управление сетью в случае отказа активного контроллера. Кстати, архитектура распределенного контура управления до сих пор остается одной из ключевых проблем в ПКС сетях. Проблемы, возникающие в физически распределенном контуре управления, включают в себя задержки на передачу сообщений в контуре управления, поддельные управляющие сообщения, несогласованные обновления и изменения маршрутизации в сети, в случае, когда пакеты все еще находятся в пути. Например, при создании избыточности для обеспечения отказоустойчивости контура управления сообщения OpenFlow, вызванные задержками в сети, могут привести к неправильным обновлениям информации в контуре передачи данных. Так что здесь есть над чем поработать.

Архитектура SDX и протокол EBGP. Организация точек обмена трафиком между автономными системами Интернета (Internet eXchange Point – IXP) в виде ПКС сети дает много преимуществ по сравнению с традиционными подходами. Прежде всего надо осознавать, что управление такой точкой обмена должно быть частью распределенного контура управления сетью. Такая организация программно-конфигурируемой точки обмена (SDX) позволит автономным системам

(АС) задавать свои политики маршрутизации в виде приложений централизованного ПКС контроллера сети точки обмена (Software Defined eXchange point – SDX), который становится арбитром глобальной политики для нескольких доменов и автономных систем. Обладая таким глобальным видением, контроллер SDX может внедрять методы масштабирования, которые могут позволить объединить большое количество политик разных АС. В результате SDX поможет разрешать проблемы междоменной маршрутизации, с которыми долгое время сталкиваются существующие точки обмена, внедряя новые политики, которые обрабатывают пакеты на более детальном уровне, при сохранении поддержки постоянной рассылки объявлений BGP маршрутов. Преимуществами SDX являются предотвращение нарушений политик обмена между доменами-участниками обмена при возникновении DDoS-атак; оптимизация маршрутов для обеспечения быстрой сходимости в сети, разгрузки каналов передачи данных, анализ трафика на middlebox'ах, инженерия трафика в контуре передачи данных; применение пиринга в зависимости от используемых приложений; дистанционное управление выбором маршрута BGP и балансировкой нагрузки в WAN сетях при передаче разнородного трафика, чувствительного к производительности.

Еще одна проблема, связанная с SDX и мультидоменными ПКС сетями, – это обеспечение соблюдения определенной политики в разных доменах. Поскольку ПКС контроллер в одном домене не может определять и контролировать политики в других доменах, сетевые операторы не могут обеспечить, чтобы их собственные политики применялись в доменах, внешних по отношению к их собственным. Поэтому проблемы проверки политики в SDX и мультидоменных ПКС сетях предоставляют обширную область для исследований, актуальную для операторов связи. Как уже было отмечено, при рассмотрении иерархических периферийных вычислений необходимо, чтобы протокол, который мы назвали Extended BGP (EBGP), разрешал АС объявлять (публиковать) другим АС, какие сервисы доступны в этой АС и в соответствии с каким SLA. АС-заявитель должен предоставлять для каждого анонсируемого сервиса как минимум следующую информацию: политику расчета стоимости оплаты, SLA, QoS, гарантированную задержку доступа, минимальную ожидаемую производительность, через какой шлюз АС-заявителя этот сервис доступен. Здесь термин «сервис» можно толковать широко: это предложение некоторых ресурсов, инфраструктуры как услуги и различных виртуализированных сервисов. Ключевым моментом является то, что АС-заявитель гарантирует условия SLA, такие как: доступность услуги, задержка доступа, джиттер, производительность. Кроме того, для поддержки гибкого сквозного управления и согласования ресурсов и сервисов в средах с несколькими АС необходима общая точка для обмена информацией о доступных ресурсах. Принципы организации такой точки и ее функционирование – самостоятельная проблема в рамках проблематики организации SDX.

Виртуализированные сервисы: интероперабельность и управление. Существующие подходы к организации управления и поддержки (MANO) сервисов, определенные ETSI, не предназначены для использования в мульти-доменных средах. Однако важным требованием гетерогенной среды иерархических периферийных вычислений является обеспечение совместимости между виртуализированными сервисами разных поставщиков при их соединении в цепочки на периферии сетевых доменов разных операторов. Объединение виртуализированных сервисов, раз-

мещенных на периферии разных сетевых доменов и от разных поставщиков, в единый сервис невозможно без четко определенных интерфейсов, в первую очередь потому, что нет общепринятой модели данных для реализации дескрипторов сервисов. Возможными решениями могут быть концепции SOA [32] и TOSCA [29]. Это еще одна проблема, требующая дополнительных исследований. Другой важной проблемой виртуализации сервисов в распределенной мультидоменной среде является поддержка функций конфигурирования, определения работоспособности, производительности, билинга и безопасности. Например, вопросы билинга и управления учетом потребления ресурсов сети и выставления счетов по-прежнему полностью игнорируется почти во всех системах, в то же время средства и методы управления безопасностью и мониторинг производительности до сих пор имеют существенные ограничения [36]. Вообще говоря, для распределенной мультидоменной среды все еще нет системы управления жизненным циклом виртуализированного сервиса.

Оркестровка ресурсов, их инициализация и мониторинг. Поскольку серверы, включая их ограниченные объемы основной памяти, вычислительные ресурсы и ресурсы внешней памяти, могут быть распределены по периферия разных доменов, а пропускная способность междоменной связи также ограничена, то управление этими ресурсами должно отличаться динамизмом, масштабируемостью и быть автоматизированным, чтобы добиться нужного экономического эффекта. Здесь можно выделить три проблемы. Это а) неопределенность задержки от и до «точки присутствия» (PoP) виртуализированного сервиса, б) управление размещением сервисов и в) динамическое управление ресурсами. Централизованный подход к управлению виртуализированными сервисами и их оркестровка, предлагаемые ETSI, налагают ограничения на масштабируемость, что особенно проблематично для услуг в мультидоменных средах из-за накладных расходов на передачу данных и задержек процессов на обработку. В результате возможные направления исследования в этом направлении включают в себя разработку эффективных механизмов мониторинга, которые лучше реагируют на динамику запросов и изменяющиеся требования в обслуживании, лучше учитывают задержки на распространение информации об изменениях в конфигурациях ресурсов, а также предоставляют информацию, необходимую для динамических изменений конфигураций, распределенных объектов. Здесь, по-видимому, нужны будут легкие коммуникационные протоколы для оптимизации использования ресурсов и повышения производительности услуг. SDN и NFV очень хорошо дополняют друг друга, при этом динамизм сетей и сервисов, их изменчивость должны быть хорошо наблюдаемы и хорошо управляемы. Следовательно, кроме традиционного управления виртуализированными вычислительными ресурсами и сервисами, должны быть разработаны дополнительные подходы к управлению и мониторингу в случае совместного использования SDN&NFV. Таким образом, управленческие решения, которые служат для объединения SDN и NFV, являются ключевыми областями исследований.

Управление качеством обслуживания (QoS) и отказоустойчивость. Поддержание необходимых уровней качества обслуживания и отказоустойчивости является важной проблемой. Иерархические периферийные вычисления в первую очередь предназначены для приложений реального времени, поэтому отказоустойчивость должна быть проактивной и должно быть реализовано автоматическое восстановление работы контура управления и контура передачи данных после сбоев.

Устройства на периферии не должны быть перегружены, чтобы поддерживать минимально необходимый уровень качества обслуживания. Следовательно, должен быть реализован надлежащий механизм мониторинга, который контролирует использование периферийных узлов в пиковые часы, тем самым облегчая гибкое распределение и планирование задач. Еще одна проблема в обеспечении качества обслуживания при периферийных вычислениях заключается в том, что в совместной работе участвуют объекты из периферий нескольких, разных сетевых доменов. Например, такой сценарий может возникать в случае сети доставки контента или когда пользователь перемещается из зоны периферии одного домена в периферийную зону другого. В этом случае пользовательские данные должны быть доступны в обеих периферийных зонах. Решением этой проблемы является совместное кэширование пользовательских данных на периферии взаимодействующих сетей. Однако это вызывает рост трафика между взаимодействующими сетями. Следовательно, должны быть разработаны оптимальные стратегии размещения данных и их репликаций, которые уменьшают задержки и собственно трафик до минимально допустимых пределов, определяемых требуемым качеством обслуживания. Сложной проблемой, связанной с качеством обслуживания, является поддержание пропускной способности сети на требуемом уровне [37].

Фильтрация контекста данных. Фильтрация контекста данных – это предварительная обработка данных на периферии сети, предшествующая их дальнейшей передаче. Выше мы приводили несколько примеров, когда устройства для «интернета вещей» или пользовательские устройства в научных экспериментах генерируют огромное количество данных. Прокачка этого объема данных через взаимодействующие сети может привести к перегрузкам в них и перегрузкам ЦОДов на периферии взаимодействующих сетей. Однако фильтрация данных вызывает несколько проблем. Если будет отфильтровано слишком много данных, это может привести к потере некоторой полезной информации, что приведет к снижению точности данных. Если данные будут отфильтрованы слабо, нежелательные данные также могут быть отправлены дальше, вызывая дополнительную нагрузку на ресурсы взаимодействующих сетей.

Список литературы / References

[1] Bilal K., et al., “Trends and challenges in cloud datacenters”, *IEEE Cloud Computing*, **1**:1 (2014), 10–20.

[2] Bilal K., et al., “A taxonomy and survey on Green Data Center Networks”, *Future Generation Computer Systems*, **36** (2014), 189–208.

[3] Bilal K., Khan S.U., Zomaya A.Y., “Green Data Center Networks: Challenges and Opportunities”, *IEEE Conference on Frontiers of Information Technology*, IEEE, 2013, 229–234.

[4] Bilal K., et al., “A survey on green communications using adaptive link rate”, *Cluster Computing*, **16**:3 (2013), 575–589.

[5] Chen Zhuo, et al., “Early implementation experience with wearable cognitive assistance applications”, *Proceedings of the 2015 workshop on Wearable Systems and Applications*, ACM, 2015, 33–38.

[6] Shi Weisong, et al., “Edge computing: Vision and challenges”, *IEEE Internet of Things Journal*, **3**:5 (2016), 637–646.

[7] "Cisco Global Cloud Index: Forecast and Methodology, 2016-2021", https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html.

[8] "4 Vs of Big Data", http://www.ibmbigdatahub.com/sites/default/files/infographic_file/4-Vs-of-big-data.jpg.

[9] Bonorni F., et al., "Fog computing and its role in the internet of things", Proceedings of the first edition of the MCC workshop T Mobile cloud computing (Helsinki, Finland), 2012, 13–16.

[10] Brown B., "Microsoft researcher: Why Micro Datacenters really matter to mobile's future", http://www.networkworld.com/article/2979570/cloud-computing/microsoft-researcher-why-micro-datacenters-really-matter-to-mobiles-future.html.

[11] Satyanarayanan M., et al., "The Case for VM-Based Cloudlets in Mobile Computing", *IEEE Pervasive Computing*, **8**:4 (2009), 14 – 23.

[12] Aazam M., Huh E., "Dynamic resource provisioning through fog micro datacenter", *The 12th IEEE International Workshop on Managing Ubiquitous Communications and Services*, 2015, 105–110.

[13] Aazam M., Huh E., "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT", *The 29th IEEE International Conference on Advanced Information Networking and Applications (AINA-15)*, IEEE, 2015, 687–694.

[14] Jararweh Y., et al., "The future of mobile cloud computing: Integrating cloudlets and Mobile Edge Computing", *2016 23rd International Conference on Telecommunications (ICT)*, IEEE, 2016.

[15] "Green Clouds", http://www.greenclouds.in/views-and-resources/high-performance-websites/.

[16] Giordano A., Spezzano G., Vinci A., "Smart agents and fog computing for smart city applications", *International Conference on Smart Cities*, Springer, 2016, 137–146.

[17] Cortes R., et al., "Stream processing of healthcare sensor data: studying user traces to identify challenges from a big data perspective", *Procedia Computer Science*, **52** (2015), 1004–1009.

[18] Costenaro D., Duer A., "The megawatts behind your megabytes: going from data-center to desktop", *2012 ACEEE Summer Study on Energy Efficiency in Buildings*, 2012.

[19] "Cisco Data in Motion", https://www.cisco.com/c/m/en_us/solutions/data-center-virtualization/data-motion.html.

[20] "MicroDC Solution", https://actfornet.com/HUAWEI_CLOUD_COMPUTING/Huawei%20MicroDC%20Brochure.pdf.

[21] "Mobile Edge Computing", http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing.

[22] "Akamai", https://blogs.akamai.com/2016/07/portugal-france-sets-live-sports\discretionary{-}{}{}streaming-record-on-akamai.html.

[23] Wang Meisong, et al., "An overview of cloud based content delivery networks: research dimensions and state-of-the-art", *Transactions on Large-Scale Data-and Knowledge-Centered Systems XX*, Springer, 2015, 131–158.

[24] Chu Weibo, et al., "Network delay guarantee for differentiated services in content-centric networking", *Computer Communications*, **76** (2016), 54–66.

[25] Wang Rui, et al., "Mobility-aware caching for content-centric wireless networks: Modeling and methodology", *IEEE Communications Magazine*, **54**:8 (2016), 77–83.

[26] Ahmed Syed Hassan, Bouk Safdar Hussain, Kim Dongkyun, *Content-Centric Networks: An Overview, Applications and Research Challenges*, Springer, 2016, 108 pp.

[27] Gao Ying, et al., *Are cloudlets necessary?*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA, 2015, Tech. Rep. CMU-CS-15-139.

[28] Bilal K., et al., "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers", *Computer Networks*, **130** (2018), 94–120.

[29] "TOSCA Simple Profile in YAML Version 1.2", http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.2/csprd01/TOSCA-Simple-Profile-YAML-v1.2-csprd01.pdf.

[30] Nygren E., Sitaraman R.K., Sun J., "The akamai network: a platform for high-performance internet applications", *ACM SIGOPS Operating Systems Review*, **44**:3 (2010), 2–19.

[31] "Data Never Sleeps 2.0", https://www.domo.com/learn/data-never-sleeps-2.

[32] Erl T., *Service-oriented architecture: concepts, technology, and design*, 2005.

[33] Antonenko V., et al., "C2: General Purpose Cloud Platform with NFV Life-Cycle Management", *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, IEEE, 2017, 353–356.

[34] Petrov I. S., "Mathematical model for predicting forwarding rule counter values in SDN", *Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018 IEEE Conference of Russian*, IEEE, 2018, 1313–1317.

[35] Petrov I., Morgunova O., "Forwarding Rule Minimization for Network Statistics Analysis in SDN", *2018 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTec)*, IEEE, 2018, 1–6.

[36] Cox J.H., et al., "Advancing software-defined networks: A survey", *IEEE Access*, **5** (2017), 25487–25526.

[37] Chemeritskiy E., Stepanov E., Smeliansky R., "Managing network resources with flow (de) multiplexing protocol", *Mathematical and Computational Methods in Electrical Engineering*, Recent Advances in Electrical Engineering Series, **53**, 2015, 35–43.

Smeliansky R. L., "Hierarchical Edge Computing", *Modeling and Analysis of Information Systems*, **26**:1 (2019), 146–169.

DOI: 10.18255/1818-1015-2019-1-146-169

Abstract. The computing paradigm based on the giant-like DC is replaced by a new paradigm. The urgency of this shift is caused by the requirements of new applications that actively use video, real-time interactivity, new mobile communication technologies, which today cannot be implemented without the usage of cloud computing and virtualization based on SDN&NFV technologies. The presentation considers the requirements dictated by these applications, outlines the architecture of this new paradigm which we call Hierarchical Edge Computing (HEC). Attention is focused on the fact that all these applications are distributed, become more and more real-time applications and require guaranteed quality of service in the networking operation. The main scientific problems that need to be solved for implementing this new paradigm are discussed.

Keywords: SDN, NFV, fog computing, cloud computing, data center, mobile edge computing

On the authors:
Ruslan L. Smeliansky, Corresponding Member of Russian Academy of Sciences, professor, doctor of sciences,
orcid.org/0000-0003-2311-4513
Lomonosov Moscow State University,
1-bd. 52 Leninskie gory, Moscow, 119992, Russia, e-mail: smel@cs.msu.ru

Acknowledgments:
This work was supported by the Russian Fund of Basic Research, Grant N 18-07-01245.