



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Романов Андрей Романович

**Разработка системы обеспечения надежного и
масштабируемого виртуального сетевого сервиса в
облачной среде**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
к.ф.-м.н.
В.А. Антоненко

Москва, 2016

Аннотация

В данной работе рассматриваются проблемы организации надежной работы и масштабируемости виртуального сетевого сервиса (Virtual Network Service, VNS). Рассмотрены существующие решения организации работы виртуальных сетевых функций (Virtual Network Function, VNF).

В рамках выпускной квалификационной работы разработан программный комплекс, позволяющий в автоматическом режиме обеспечивать отказоустойчивость и масштабируемость виртуальных сетевых функций в облаке.

Оглавление

Введение	5
1 Постановка задачи	8
2 Обзор предметной области	9
2.1 Общее описание NFV	9
2.2 Архитектура ETSI NFV MANO	10
2.2.1 Менеджер виртуальной инфраструктуры	12
2.2.2 Менеджер виртуальных сетевых функций	13
2.2.3 Оркестратор виртуальных сетевых сервисов	15
3 Обзор существующих NFV платформ	17
3.1 Open Platform for NFV	17
3.2 Cloudify	18
3.3 OpenStack Tacker	20
3.4 CORD on.lab	21
3.5 OpenBaton	21
3.6 Проприетарные решения	22
4 Исследование и построение решения задачи	23
4.1 Анализ результатов обзора	23
4.2 План построения решения задачи	24

5	Описание практической части	25
5.1	Облачная платформа Cloud Conductor	25
5.1.1	Модуль мониторинга	26
5.1.2	Базовый модуль	26
6	Экспериментальные исследования	27
6.1	Эксперимент healing	27
6.1.1	Описание эксперимента	27
6.1.2	Входные данные	27
6.1.3	Ожидаемая реакция	28
6.1.4	Результаты эксперимента	28
6.2	Эксперимент scaling	28
6.2.1	Описание эксперимента	28
6.2.2	Входные данные	29
6.2.3	Легенда	29
6.2.4	Ожидаемая реакция	29
6.2.5	Результаты эксперимента	30

Введение

В современных сетях функционирует огромное количество сервисов: маршрутизация (routing), трансляция сетевых адресов (NAT), сетевой экран (firewall), туннелирование (VPN), прокси-сервер и т.д.. Многие из них реализованы в одном устройстве. Для эффективной работы сервисов нагрузку распределяют сразу на несколько таких устройств. При необходимости в новой функции требуется купить новое оборудование, которое будет обладать прочими ненужными функциями. При высокой нагрузке, устройство не будет справляться со своими задачами, а значит придется купить еще одно. В случае недостаточной нагрузки устройства будут простаивать. Здесь возникает проблема динамической масштабируемости сервиса в зависимости от его загрузки. Известно, что при наличии нескольких схожих по функциональности устройств от разных производителей, могут возникать конфликты в их работе. Таким образом, одним из решений является приобретение оборудования только от одного вендора. Со временем производитель отказывается поддерживать устаревшее оборудование. И в этом случае остается крайний вариант - приобретение новых устройств взамен устаревших.

Таким образом можно выделить ключевые проблемы организации работы сетевого сервиса:

- приобретение оборудования с избыточной функциональностью;

- требуется рассчитывать производительность сервиса исходя из максимальной возможной нагрузки;
- простаивание оборудования в случае, если нагрузка не является пиковой;
- зависимость от производителя оборудования (тех. обслуживание, устаревание оборудования, невозможность модифицировать сервис без вмешательства производителя);

Концепция виртуальных сетевых функций (Network Function Virtualization, NFV) — это молодая технология, позволяющая виртуализировать некоторые программные сервисы, которые на данный момент реализованы лишь на физических устройствах. NFV работает в рамках модели SaaS (Software as a Service, программное обеспечение как услугу) и обладает такими качествами, как:

- масштабируемость - в зависимости от загруженности сервиса будет работать тот объем инфраструктуры, который необходим для надежной работы;
- надежность - в случае сбоев в работе сервиса будут предприниматься действия по восстановлению его работы в автоматическом режиме;
- гибкость - виртуализация позволяет быстро развертывать сервисы на новой инфраструктуре;
- безопасность - данные клиентов защищены, так как программные сервисы работают изолированно друг от друга.

Организация ETSI разработала высокоуровневую архитектуру ETSI NFV Management and Orchestration (ETSI NFV MANO). Главной особенностью ар-

хитектуры является оптимальное использование инфраструктуры: она выделяется для каждой функции по запросу в необходимом количестве. Базовыми блоками, из которых строятся виртуальные сетевые сервисы, являются виртуальные сетевые функции (VNF). Платформа на базе ETSI NFV MANO умеет размещать VNF на подконтрольной инфраструктуре. В результате комбинирования блоков VNF получаются виртуальные сетевые сервисы (VNS), которыми пользуются клиенты платформы.

1 Постановка задачи

Разработать решение, управляющее жизненным циклом виртуальных сетевых сервисов, обеспечивающее их надежную работу и масштабируемость.

Решение должно удовлетворять следующим требованиям:

1. по запросу осуществлять подписку и отписку пользователей от виртуальных сетевых сервисов в рамках модели Software as a Service (SaaS);
2. при возникновении неисправности принимать меры по восстановлению корректной работы сервиса в автоматическом режиме (healing);
3. обеспечивать масштабируемость инфраструктуры сервиса в автоматическом режиме (scaling);
4. решение должно быть независимым от платформы виртуализации инфраструктуры;
5. решение должно быть согласовано с высокоуровневой архитектуры ETSI NFV MANO;
6. поддержка нескольких платформ виртуализации инфраструктуры одновременно;

2 Обзор предметной области

2.1 Общее описание NFV

В 2004 году была предложена идея организации сетевой инфраструктуры с целью снижения затрат и ускорения внедрения новых услуг. Она состояла в объединении ядра сети и сети доступа в единую платформу. Однако без виртуализации идея не получила широкого распространения.[4].

NFV предполагает использование виртуализированной инфраструктуры для функционирования услуг. Концепция предлагает использование технологий для виртуализации функций в виде составных элементов, которые могут быть связаны для создания телекоммуникационных сервисов.

Таким образом, виртуальная сетевая функция (VNF) — это описание требуемой инфраструктуры, требуемого программного обеспечения, параметров подключения пользователей к этой услуге и т.д.. Заметим, что программное обеспечение, описанное в VNF должно иметь ограниченную и законченную функциональность. Не следует виртуализировать сложное программное обеспечение, так как оно будет использовать дополнительные ресурсы инфраструктуры.

Виртуальный сетевой сервис (VNS) - это некоторое множество связанных между собой виртуальных сетевых функций. Это конечная услуга, которая будет предоставляться клиентам. Концепция предполагает внутреннее пред-

ставление VNS как произвольное непустое множество, состоящее из VNF. При этом VNF как-то связаны друг с другом.

По мнению автора, наиболее интересен случай цепочек виртуальных сетевых функций (VNF chaining). В этом случае можно считать каждую VNS как цепочку сетевых функций. Близкую аналогию можно провести с математическим понятием функции. Пусть 'x' - это входящий трафик некоторого объема. Тогда результатом работы сервиса S, состоящего из последовательной цепочки функций f1, f2, f3 будет трафик y, такой что:

$$y = f3(f2(f1(x))) = S(x)$$

В результате трафик 'x' трансформировался в трафик 'y'. S - это суперпозиция функций f1, f2, f3.

2.2 Архитектура ETSI NFV MANO

Европейский Институт Телекоммуникационных Стандартов (ETSI) в 2013 году опубликовал высокоуровневые рекомендации по разработке платформы, управляющей виртуальными сетевыми сервисами Network Function Virtualization Management and Orchestration (NFV MANO). Основные цели документы - стандартизировать интерфейсы каждого модуля в платформе.[2] Как показано на Рис. 2.1, в NFV MANO имеется 3 основных модуля:

1. менеджер виртуальной инфраструктуры (Virtualized Infrastructure Manager, VIM)
2. менеджер виртуальных сетевых функций (Virtual Network Function Manager, VNFM)
3. оркестратор виртуальных сетевых сервисов (Network Function Virtualization, NFVO)

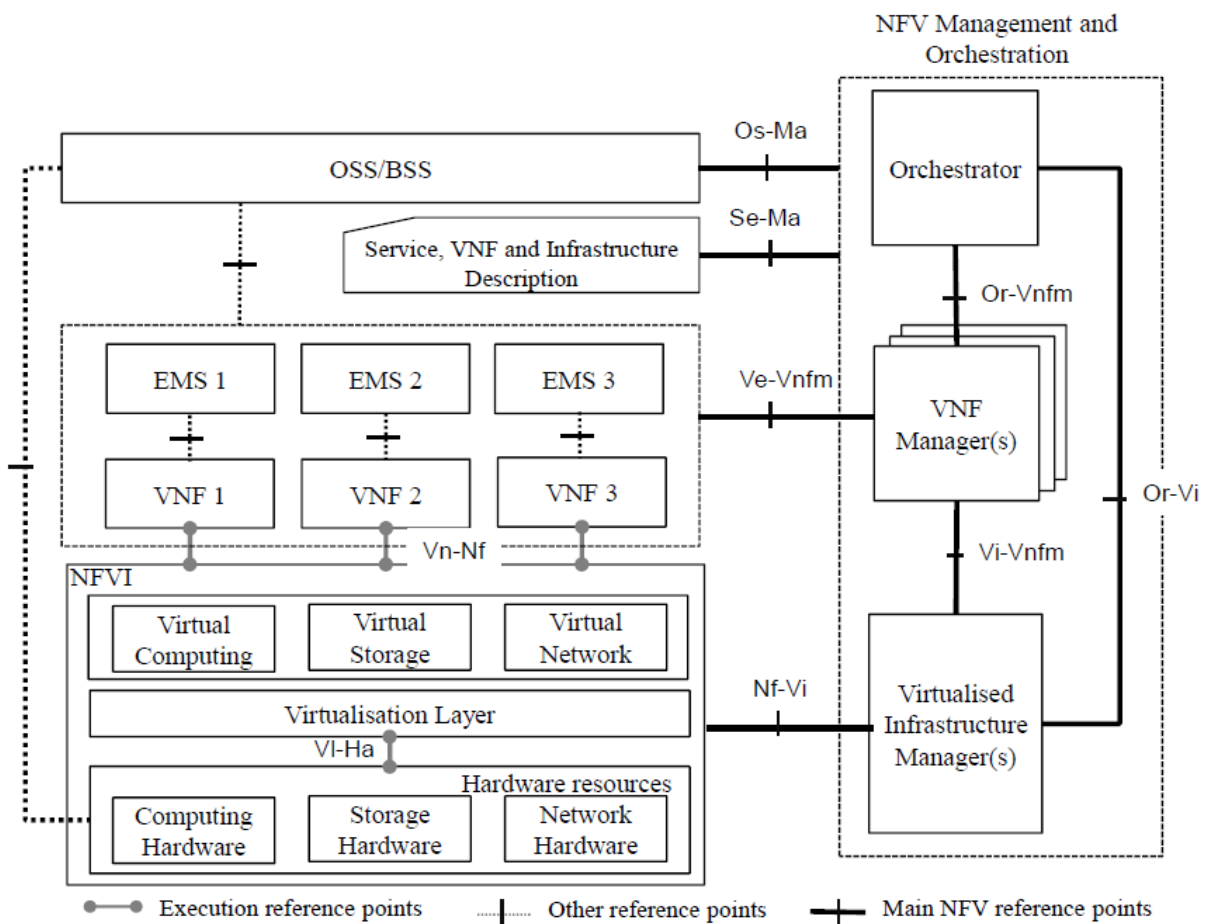


Рис. 2.1: Архитектура NFV Management and Orchestration

Каждый модуль обеспечивает свой слой виртуализации. VIM занимается виртуализацией физических ресурсов. Менеджер функций предоставляет набор функций, размещенных на виртуальных ресурсах. Оркестратор управляет сетевыми сервисами, построенными на базе виртуальных сетевых функций.

Так же в архитектуре присутствуют неосновные модули:

- описание виртуальных сетевых сервисов, функций и используемой ими инфраструктуры. В NFV MANO блоки, которые содержат такие описания, называют каталогами (catalog). В платформах, разрабатываемых на базе NFV MANO, обычно функции каталогов выполняет менеджеры соответствующего уровня (оркестратор сервисов, менеджер функций,

VIM);

- система управления элементами (Element Management System, EMS). Система управляет работой элементов экземпляра виртуальной сетевой функции, отвечает за параметры функции. Данная система взаимодействует с менеджером функций через закрытые интерфейсы. Поэтому в существующих решениях, известных автору, данный модуль включен в состав менеджера функций.

Далее рассмотрим подробнее основные модули NFV MANO.

2.2.1 Менеджер виртуальной инфраструктуры

Менеджер инфраструктуры обеспечивает виртуализацию физической инфраструктуры в рамках одного домена. NFV MANO предполагает использование нескольких менеджеров инфраструктуры в одном домене. В задачи модуля входит:

- управление полным жизненным циклом виртуальных ресурсов в рамках одного домена
 - управление вычислительными ресурсами (computing resource) и хранилищами (storage);
 - управление сетевыми ресурсами (networking resource), то есть управление коммутаторами, роутерами, сетевая настройка;
 - остальные задачи гипервизора (абстракция физической инфраструктуры, эффективное отображение ресурсов на их виртуальные аналоги и т.д.);
- иметь полную информацию о доступных физических ресурсах и о запущенной виртуальной инфраструктуре;

- мониторинг за состоянием виртуальных ресурсов, обнаружение отказов оборудования, виртуальных машин, программного обеспечения;
- оповещать остальные модули о смене состояния виртуальной инфраструктуры;
- предоставление интерфейса для использования виртуальной инфраструктуры и для мониторинга физической инфраструктуры.

Подробнее о функциях VIM, и его спецификациях можно прочитать в стандарте [1].

2.2.2 Менеджер виртуальных сетевых функций

Менеджер виртуальных сетевых функций (VNFM) — это основной модуль архитектуры NFV MANO, ответственный за полный жизненный цикл виртуальных сетевых функций. Архитектура предполагает возможность наличие нескольких менеджеров функций. Однако на момент написания работы остается неясным, необходимо ли наличие нескольких модулей.

В NFV MANO следует отличать два понятия: описание виртуальной сетевой функции (VNF description) и экземпляр виртуальной сетевой функции (VNF instance). Когда говорят про сетевую функцию, чаще всего имеют ввиду ее спецификации, то есть ее описание. Экземпляр VNF - это виртуальная инфраструктура, которая уже размещена поверх физической, и полностью удовлетворяют своему описанию (спецификациям). Таким образом, для каждой VNF существует единственное описание и множество ее экземпляров. Все экземпляры функции независимы друг от друга. В общем случае, они могут быть размещены в разных доменах (в разных VIM).

В задачи модуля входит:

- регистрация и удаление VNF(в случае, если менеджер функций управляет сразу несколькими функциями);
- владение полной информацией о всех спецификациях функции
 - топология инфраструктуры, которую необходимо разместить для работы функции;
 - входные параметры функции (TODO: привести пример);
 - исчерпывающая информация о программном обеспечении виртуальных машин;
 - описание событий, по которым можно судить о состоянии функции (например, в ситуации, когда инфраструктура экземпляра функции недоступна, можно считать, что функция не работает)
 - описание триггеров на вышеуказанные события (перезапустить виртуальную машину в случае, если она не отвечает на команду ping в течении 5 секунд)
- управление жизненным циклом виртуальной функции
- отслеживание неисправностей в работе программного обеспечения функции;
- реагировать на неисправности в работе VNF в соответствии с ее описанием (применить горизонтальное масштабирование в ответ на событие о недостатке производительности виртуальной машины);
- после размещения виртуальной инфраструктуры инициализировать ее, установить и настроить все необходимое для работы функции программного обеспечение;

- обновление программного обеспечения уже размещенных виртуальных функций;
- оповещать остальные модули о событиях, связанных с работой VNF;

Более подробную спецификацию менеджера виртуальных функций можно прочитать в стандарте [1].

2.2.3 Оркестратор виртуальных сетевых сервисов

Оркестратор сетевых сервисов решает две основные задачи:

- оркестрация ресурсов между несколькими менеджерами инфраструктуры, резервация ресурсов;
- управление жизненным циклом виртуальных сетевых сервисов;

Задача управления виртуальными сервисами нетривиальна. Она включает в себя множество подзадач:

- предоставление интерфейса создания, удаления, изменения, обновления VNS;
- авторизация для управления сетевыми сервисами, разграничение прав доступа;
- синхронизация работы с менеджерами функций;
- отслеживание неисправностей в работе сервисов;
- реакция на события, связанные с изменением состояния сервиса;
- оповещение остальных модулей об изменении состояния сервисов;

- резервация ресурсов под сервисы с помощью модуля VIM;

Более конкретный стандарт архитектуры NFV оркестратора находится в разработке, поэтому информацию о менеджере виртуальных сетевых сервисов можно получить, например, в высокоуровневом стандарте [1].

3 Обзор существующих NFV платформ

3.1 Open Platform for NFV

Open Platform for NFV (OPNFV) - это платформа с открытым исходным кодом, на базе которой можно создавать компоненты идеологии NFV. Проект OPNFV, используя архитектуру ETSI NFV MANO, фокусируется на разработке менеджера инфраструктуры (NFVI) архитектуры NFV MANO.[6]

Основные цели проекта:

- разработка интегрированной и протестированной открытой платформы, которая сможет быть использована для построения NFV, ускорения внедрения новых продуктов и сервисов;
- привлечение заинтересованных лиц со стороны конечных заказчиков для удовлетворения требований пользовательского сообщества.
- создать экосистему NFV решений, основанную на открытых стандартах и программном обеспечении, удовлетворяющую требованиям конечных пользователей;
- продвигать OPNFV как предпочтительную платформу и сообщество для создания NFV решений с открытым кодом.

OPNFV стремится участвовать в смежных открытых проектах, которые могут быть использованы в OPNFV, обеспечить целостность, производительность и функциональную совместимость компонентов. OPNFV активно взаимодействует с открытыми проектами: OpenStack, KVM, Open vSwitch, OpenDyalight, ONOS, Open Contrail, ETSI, IETF. Сообщество состоит из более чем 60 компаний, начиная с производителей оборудования и заканчивая поставщиками SDN и NFV решений.

Первый релиз (Arno) состоялся в июня 2015 году и какой-либо функциональности в себе не нес. Вторая версия проекта OPNFV (Brahmaputra) вышла 1 марта 2016 года. По словам сообщества, теперь платформа готова для проведения лабораторных тестов.[7]

Как уже было отмечено, OPNFV - это база для реализации продуктов на базе NFV MANO. В данной платформе разрабатывается лишь модуль NFVI, отвечающий за виртуальные ресурсы. Задачи по масштабируемости и отказоустойчивости здесь выполняются только на уровне виртуальных и физических ресурсов.

3.2 Cloudify

Cloudify - это платформа с открытым исходным кодом. Cloudify архитектурно состоит из основного модуля, называемого Cloudify Manager VM, и Cloudify агентов, установленных на подконтрольных виртуальных машинах.

Cloudify Manager VM исполняет роли сразу двух основных модулей - это VNFM и NFVO. Таким образом, указанный модуль выполняет множество задач:

- регистрация новых виртуальных функций. Описание функций представляется в формате собственной разработки, называемый blueprints.

Он основан на стандарте описания функций TOSCA (формат, основанный на YAML);

- размещение инфраструктуры VNF, используя плагины к существующим платформам виртуализации ресурсов (поддерживаются Openstack, VMware);
- инициализация инфраструктуры функций, используя программы-агенты на подконтрольных виртуальных машинах;
- мониторинг изменения состояния виртуальных функций с помощью агентов;
- запуск триггеров из описания функции;

Cloudify агенты ответственны за выполнения команд Cloudify Manager VM. Различают агентов со стороны Cloudify менеджера (manager side agents) и со стороны виртуальной сетевой функции (application side agents). Агенты менеджера устанавливаются вместе с операционной системой виртуальной машины и выполняют следующие служебные задачи: создание виртуальной машины, привязка внешнего ip-адреса и т.д.. Агенты виртуальной функции являются опцией (устанавливаются, если стоит соответствующая запись в описании функции). Задачи, выполняемые агентами функции, должны присутствовать в описании функции.[8]

Изучение платформы Cloudify показало, что в действительности полной автоматизации процесса мониторинга и срабатывание триггеров еще не достигнуто. После размещения функции требуется часть настроек произвести в ручном режиме. TODO: изучить срабатывание триггеров, а именно scaling и healing.

3.3 OpenStack Tacker

Openstack Tacker - это проект с открытым исходным кодом. Использует разработки проекта OPNFV. Основной целью проекта является реализация основных блоков ETSI NFV MANO (VNF-Manager и VNF-Orchestrator) в виде плагина для платформы облачной виртуализации Openstack. Tacker реализует управление виртуальными функциями и оркестрацию сетевых сервисов. Рассмотрим основную функциональность базовых блоков архитектуры ETSI NFV MANO в рамках проекта Tacker. Основные задачи, выполняемые блоком VNF-Manager:

- хранилище всех виртуальных функций, доступных системе;
- управление полным жизненным циклом каждой виртуальной функции (размещение, инициализация, масштабирование, остановка, удаление);
- мониторинг за размещенными виртуальными функциями. Основные параметры мониторинга: производительность и отказоустойчивость работы функции;
- автоматическое восстановление работы функции в случае ее полного или частичного отказа в предоставлении услуги по заданным политикам;
- облегчение первоначальной настройки виртуальной сетевой функции;

Задачи, выполняемые блоком VNF-Orchestrator:

- использования шаблонов при управлении сетевыми сервисами, комбинируя различные виртуальные функции между собой;
- обеспечение эффективного размещения виртуальных функций;

- создание цепочек виртуальных сетевых функций (сетевые сервисы);
- контроль за выделение ресурсов с помощью блока VIM;
- оркестрация виртуальных функций на множестве различных блоков VIM.

На текущий момент возможности Tasker реализованы только командном интерфейсе и не доступны в графическом интерфейсе Horizon платформы Openstack.[9] Восстановление работы функции и расширение инфраструктуры функции доступно только в ручном режиме.

3.4 CORD on.lab

3.5 OpenBaton

OpenBaton - проект с открытым исходным кодом, реализующий архитектуру ETSI NFV MANO. Основными модулями платформы являются:

- оркестратор сетевых сервисов NFVO;
- менеджер виртуальных сетевых функций VNFM;

Основным модулем, над которым ведется разработка - это NFVO. В нем содержится основная функциональность по размещению функций, слежению за их состоянием, восстановлению из аварийного состояния и масштабированию. VNFM - является заменяемым модулем: возможно использование модуля управления функциями собственной разработки. При этом с OpenBaton поставляются библиотеки, позволяющие упростить разработку и интегрирование собственного VNFM с оркестратором.

OpenBaton независима от платформы виртуализации ресурсов. На текущий момент разработан только плагин под платформу Openstack. Разработчиками заявлена поддержка нескольких VIM. В OpenBaton для включения функции мониторинга необходимо дополнительно установить Zabbix сервер (о поддержке других решений по слежению за виртуальными машинами автору не известно).

На момент написания работы в OpenBaton идет разработка следующей функциональности: развертывания дополнительной инфраструктуры и разнообразные улучшения в blueprints. Из этого следует, что о реализации автоматического масштабирования и восстановления функции речи пока не идет.

Для реализации собственных виртуальных сетевых сервисов OpenBaton предлагает либо реализовать менеджер функций собственной разработки, либо привести описания функции через VNFPackage. VNFPackage - это описание функции на основе формата YAML, который содержит все необходимое описание о виртуальной функции.

3.6 Проприетарные решения

Автору не известны проприетарные решения, реализующие концепцию NFV. Наиболее известные решения Microsoft Azure и VMware vSphere работают в рамках модели IaaS (Infrastructure as a Service, инфраструктура как услуга). Указанные платформы можно использовать только в качестве менеджера инфраструктуры в рамках ETSI NFV MANO.

4 Исследование и построение решения задачи

4.1 Анализ результатов обзора

Результаты обзора, приведенные в таблице 2.1, показали, что ни одно из существующих решений полностью не удовлетворяет установленным требованиям.

Платформа	ETSI NFV MANO	Независимость от платформы виртуализации ресурсов	Одновременная работа с несколькими VIM	Мониторинг состояния VNS	автоматическое срабатывание триггеров scaling, healing
OPNFV	+	+	-	?	?
Cloudify	+	+	?	+	-
Openstack Tacker	+	-	-	?	?
OpenBaton	+	+	+	+	-

Таблица 4.1: Сравнение существующих NFV решений

4.2 План построения решения задачи

Прежде чем перейти непосредственно к решению задачи необходимо понять, какие модули архитектуры ETSI NFV MANO относятся к установленным требованиям.

Оркестратор сетевых сервисов согласно ETSI NFV MANO занимается:

- подпиской и отпиской пользователей от сервиса;
- масштабируемостью инфраструктуры сервиса (scaling) в автоматическом режиме;
- восстановлением корректной работы сервиса при возникновении неисправности (healing) в автоматическом режиме.

Поддержка нескольких модулей VIM также решается на уровне оркестратора. Независимость от платформы виртуализации решается на уровне менеджера инфраструктуры (этот модуль должен использовать плагин для работы с конкретной платформой виртуализации ресурсов).

Таким образом, для решения поставленной задачи достаточно:

1. реализовать оркестратор сетевых сервисов (NFVO) согласно архитектуры ETSI NFV MANO;
2. внедрить оркестратор в платформу, удовлетворяющую следующим требованиям:
 - реализован VIM, не зависящий от платформы виртуализации ресурсов;
 - реализован модуль мониторинга за виртуальными машинами.

5 Описание практической части

В текущей главе приведено описание реализации модуля NFVO для облачной платформы Cloud Conductor. Данная платформа полностью удовлетворяет требованиям, описанным в разделе 4.2. Проект разрабатывается в отечественной организации Центр Прикладных Исследований Компьютерных Сетей (ЦПИКС).

5.1 Облачная платформа Cloud Conductor

Рассматриваемая облачная платформа ориентирована на предоставление услуг по модели IaaS. Основной задачей проекта - управление несколькими платформами виртуализации ресурсов (в частности Openstack). Взаимодействие между внутренними модулями осуществляется через библиотеку RabbitMQ (RMQ).

В рамках проекта Cloud Conductor разработаны:

- модуль мониторинга Monitoring;
- базовый модуль CORE размещения сетей виртуальных машин (тенантов).

На данный момент идет разработка NFV компонент.

5.1.1 Модуль мониторинга

Модуль мониторинга базируется на уже существующей программе, выполняющей слежение за виртуальными машинами. В качестве такой программы используется Zabbix (также возможна интеграция с другими программами мониторинга).

5.1.2 Базовый модуль

6 Экспериментальные исследования

В качестве доказательства выполнения требований 2 и 3 из раздела 1 проведем эксперименты healing и scaling, в которых покажем автоматическое восстановление работы сервиса и автоматическое расширение инфраструктуры сервиса соответственно.

6.1 Эксперимент healing

6.1.1 Описание эксперимента

Во время использования клиентом виртуального сетевого сервиса происходит отказ в инфраструктуре функции. Оркестратор должен восстановить работу сервиса.

6.1.2 Входные данные

- зарегистрирована виртуальная функция проху в менеджере функций;
- зарегистрирован виртуальной сетевой сервис прокси, состоящий из одной функции проху;

- размещена виртуальная машина клиента в облаке (напомним, что платформу Cloud Conductor позволяет размещать виртуальные машины клиентов в своем облаке);
- виртуальная машина клиента подключена к сервису прокси.

6.1.3 Ожидаемая реакция

В результате недоступности одной из виртуальных машин, модуль мониторинга генерирует событие, соответствующее отказу соединения между виртуальной машиной клиента и экземпляром сервиса, и отправляет его оркестратору сетевых сервисов. Оркестратор, получив событие об отсутствии соединения между клиентом и сервисом, решает перезапустить неправильно работающие сетевые интерфейсы виртуальных машин. Таких интерфейсов два: у виртуальной машины клиента и у виртуальной машины сетевой функции. Запрос на перезапуск интерфейсов отправляется в менеджер инфраструктуры. После получения ответа о успешном перезапуске интерфейсов оркестратором клиент продолжает потреблять сервис.

6.1.4 Результаты эксперимента

TODO

6.2 Эксперимент scaling

6.2.1 Описание эксперимента

В результате высокой нагрузки инфраструктура сервиса оказывается перегружена. Поэтому клиенты испытывают дискомфорт при потреблении сер-

виса (задержки). Оркестратор должен увеличить объем ресурсов, выделяемых для работы сервиса, чтобы исправить ситуацию.

6.2.2 Входные данные

- оркестратор работает с двумя менеджерами инфраструктуры, каждый из которых управляет ресурсами подконтрольного ему центра обработки данных (ЦОД).
- зарегистрирована виртуальная функция ргоху в менеджере функций;
- зарегистрирован виртуальной сетевой сервис прокси, состоящий из одной функции ргоху;
- размещена виртуальная машина клиента в облаке;
- экземпляр функции, использующийся клиентом, размещен на первом ЦОД;
- виртуальная машина клиента подключена к сервису прокси.

6.2.3 Ожидаемая реакция

При обнаружении перегрузки одной из виртуальных машин сервиса, модуль мониторинга генерирует соответствующее событие и отправляет его менеджеру функций. Получив событие о некорректной работе виртуальной машины из функции ргоху, менеджер считывает описание функции. Согласно описанию менеджер делает запрос в оркестратор на расширение инфраструктуры сервисного тенанта. Оркестратор оценивает запрос менеджера на инфраструктуру и оставшийся запас ресурсов в первом ЦОД. Ресурсов в первом

ЦОД оказывается недостаточно, но во втором ЦОД их хватает для удовлетворения запроса. Поэтому оркестратор решает переместить тенант с функцией с первого ЦОД на второй. После получения ответа об успешном размещении функции клиент продолжает потреблять сервис.

6.2.4 Результаты эксперимента

TODO

Заключение

Обзоры существующих платформ виртуализации сервисов показали, что ни одно из существующих решений не удовлетворяет поставленной задаче. В рамках данной работы разработано модуль для существующей NFV платформы, позволяющее автоматизировать процессы восстановление и расширение сетевых сервисов. Решение позволяет осуществлять управление подключением пользователей к сервисам.

Литература

- 1 Network Functions Virtualisation (NFV); Management and Orchestration. URL:
- 2 ETSI NFV Management and Orchestration (MANO) простым языком. URL: <https://sdnblog.ru/etsi-nfv-mano-beginners-tutorial/> (дата обращения 01.04.2016)
- 3 Network Functions Virtualisation (NFV); use cases. URL: <http://www.etsi.org/technologies-clusters/technologies/nfv> (дата обращения 01.04.2016)
- 4 NFV для корпоративных сервисов - № 12, 2014. URL: <http://www.osp.ru/lan/2014/12/13044225> (дата обращения 01.04.2016)
- 5 NFV виртуализация сетевых функций. URL: <http://sci-article.ru/stat.php?i=1455156066> (дата обращения 01.04.2016)
- 6 Open Platform for NFV (OPNFV). URL: <https://www.opnfv.org> (дата обращения 01.04.2016)
- 7 Чем занимается сообщество OPNFV? URL: <https://sdnblog.ru/who-is-opnfv/> (01.04.2016)
- 8 Cloudify Overview. URL: <http://getcloudify.org/guide/3.1/overview-architecture.html> (дата обращения 01.04.2016)

- 9 Tacker. URL: <https://wiki.opfirewallenstack.org/wiki/Tacker> (дата обращения 01.04.2016)