

OPENNEBULA В SDN СЕТЯХ ДЛЯ ПОСТРОЕНИЯ IaaS СЕРВИСОВ

С.В.Малахов, Т.В.Исаков

Поволжский государственный университет телекоммуникаций и информатики

В статье определяется производительность узлов вычислительного кластера с использованием протокола OpenFlow и без него. Производится установка контроллера FloodLight, развёртывание вычислительного кластера средствами платформы OpenNebula. Подсчитывается объём трафика, который был сгенерирован, а также средняя скорость передачи информации между узлами кластера.

Введение

Скачок в развитии высокоскоростных технологий передачи данных и вычислительных мощностей дали толчок к развитию облачных технологий. Однако на сегодняшний день устаревшие сетевые технологии не только усложняют обслуживание и тормозят рост сервисов на базе облака, но и являются «бутылочным горлышком» для центров обработки данных (ЦОД), Сетей хранения данных (СХД) и массовых пользователей сетью интернет. Большинство ключевых протоколов были разработаны на заре становления Интернета и Ethernet технологий. Тогда было невозможно предвидеть скорости и объёмы передаваемых данных, достигнутых сегодня. Объём трафика и его виды растут в геометрической прогрессии.

Теоретическая часть

Начнём с самого главного - с определения облачных вычислений.

Облачные вычисления – это модель предоставления удобного сетевого доступа к совместно используемому набору настраиваемых вычислительных ресурсов (серверам, приложениям, устройствам хранения данных и сетям передачи данных— как вместе, так и по отдельности), которые потребитель может легко задействовать в любом объёме для удовлетворения своих потребностей. Главная цель, которую преследуют облачные вычисления — это повышение доступности вычислительных ресурсов. На основании этой цели выделяют три модели обслуживания. [1]

Модели обслуживания:

- SaaS (Software as a Service) – ПО как услуга;
- PaaS (Platform as a Service) – Платформа как услуга;
- IaaS (Infrastructure as a Service) – Инфраструктура как услуга. [2]

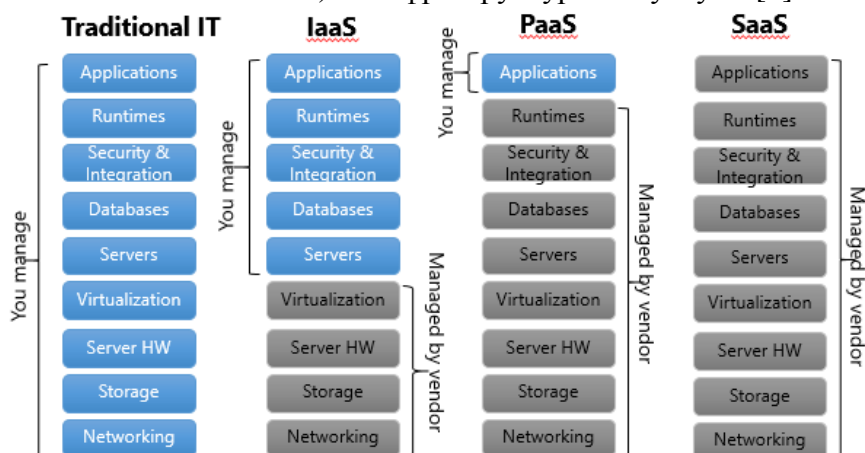


Рисунок 1 – Сравнительная характеристика моделей обслуживания

В данной работе мы рассмотрим только третью модель обслуживания. Инфраструктура как услуга (IaaS) – предоставление вычислительных ресурсов, на которых потребитель может развернуть и запустить необходимое ему ПО, включающее в

себя операционные системы и приложения. В рамках данной модели потребитель не управляет и не контролирует лежащую в основе физическую инфраструктуру.

Программно-конфигурируемые Сети (Software Defined Networks) – это активно развивающаяся архитектура сети, в которой функция управления сетью полностью программируема, а также разделена с функцией передачи данных. Основной целью SDN является возможность управления маршрутизаторами и коммутаторами, не изменяя существующего сетевого оборудования. Это достигается за счёт внедрения специального программного обеспечения (FloodLight, NOX, POX), которое работает на отдельном компьютере, под контролем администратора сети. Протокол OpenFlow позволяет управлять сетевыми коммутаторами и маршрутизаторами, благодаря чему специалисты тратят меньше времени на начальные настройки сети. Протокол OpenFlow решает также проблему зависимости от сетевого оборудования какого-либо конкретного поставщика, поскольку SDN использует общие абстракции для пересылки пакетов, которые сетевая операционная система использует для управления сетевыми коммутаторами. [3]

В архитектуре SDN выделяют три уровня:

- инфраструктурный уровень;
- уровень управления;
- уровень сетевых приложений.

OpenNebula - открытая платформа, позволяющая организовать управление cloud-инфраструктурой и виртуальным окружением. Доступно взаимодействие таких атрибутов, как сеть, хранилище, технология виртуализации, средства мониторинга, средства безопасности. [4]

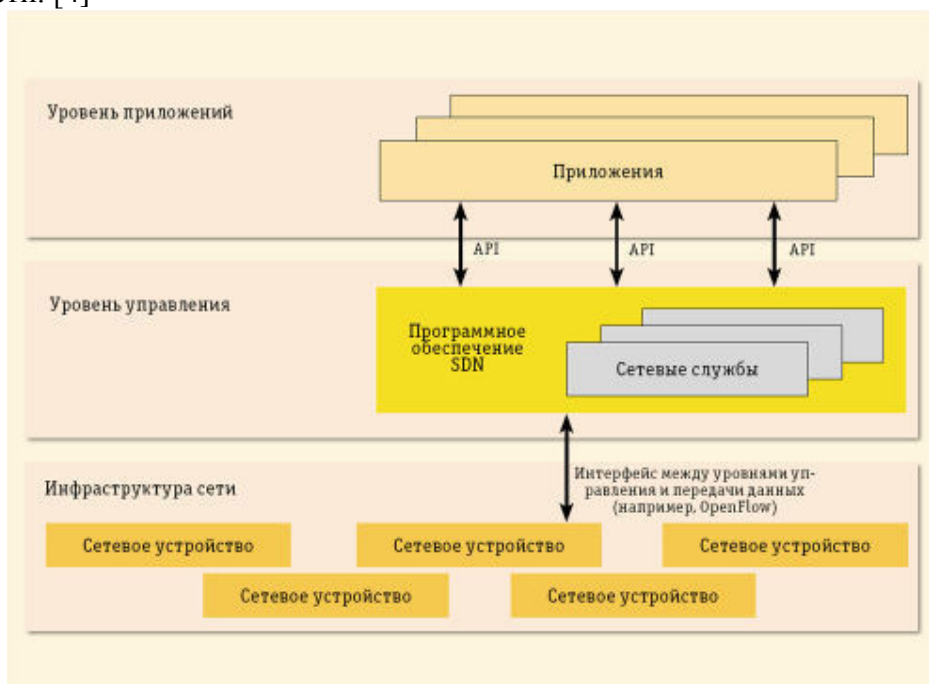


Рисунок 2 – Архитектура SDN сети

Система позволяет создавать Cloud-окружения трех типов:

- приватные Cloud-системы, доступны только внутри организации, выполнены на собственных мощностях;
- публичные Cloud-системы, работающие в инфраструктуре внешних сервисов. Для доступа используются различные API (OGF OCCI, vCloud);
- гибридные Cloud-системы, сочетают в себе элементы приватных и публичных cloud-систем.

В данной работе мы будем создавать Cloud-окружение первого типа.

Реализационная часть

Для исследования поведения пакетов было решено построить экспериментальный сегмент сети в составе:

- коммутатор OpenFlow HP 3500yl;
- 7 серверов:
 1. Server - центральный сервер (Xeon E5645 2,4ГГц, ОЗУ 32ГБ).
 2. Node-1 и Node-2 (Xeon E5645 2,4ГГц, ОЗУ 32ГБ).
 3. Node-3 и Node-4 (Core i5 2400 3ГГц, ОЗУ 32ГБ).
 4. Node-5 и Node-6 (Core i5 2320 3ГГц, ОЗУ 32ГБ).
- каналы связи между узлами сети 1000Мб/сек.

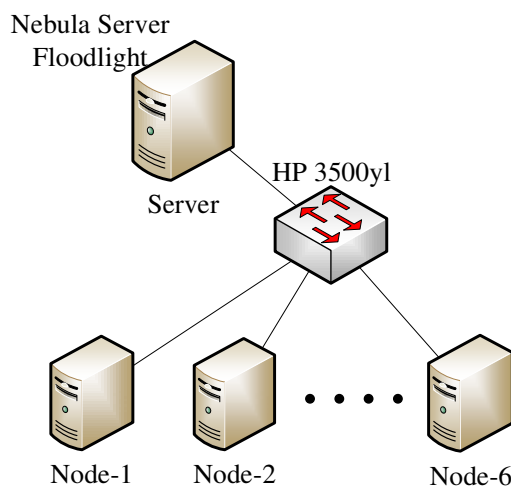


Рисунок 3 – Экспериментальный сегмент сети

Для начала нам необходимо установить контроллер FloodLight на Server. FloodLight написан на Java и, следовательно, работает в JVM. Перед установкой контроллера сначала потребуется установить, необходимые для работы, библиотеки:

1) JDK (Java Development Kit) — бесплатно распространяемый компанией Oracle Corporation (ранее Sun Microsystems) комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE) [5];

2) Ant - утилита для автоматизации процесса сборки программного продукта, является платформонезависимым аналогом утилиты make.

Для установки контроллера необходимо выполнить следующие команды:

а) команду «`sudo apt-get install build-essential default-jdk ant python-dev git`», с помощью этой команды устанавливаем необходимые библиотеки и утилиты;

б) команду «`git clone git://github.com/floodlight/floodlight.git && cd floodlight && ant`», для загрузки ветки из git, и выполнения сборки проекта.

Результатом сборки стал исполняемый jar файл floodlight.jar. Запустить контроллер можно командой «`java -jartarget/floodlight.jar`»

Далее требуется установить платформу OpenNebula, благодаря которой возможно развернуть виртуальный вычислительный кластер.

Установку OpenNebula осуществляем с помощью утилиты «`sudo dpkg -i Ubuntu-12.04-opennebula_x.x.x_amd64.deb`». В процессе установки создается учётная запись пользователя и группа oneadmin с домашним каталогом /var/lib/one/. Установка на управляемые хосты производится таким же образом. Настройки OpenNebula Sunstone располагаются в файле /etc/one/sunstone-server.conf. Далее запускаем сервис командой «`oneadmin$ sunstone-server start`».

В результате установки OpenNebula мы имеем вычислительный кластер.

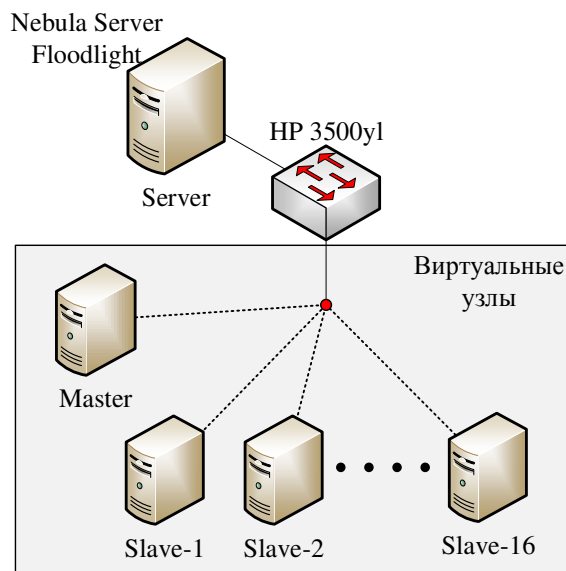


Рисунок 4 - Виртуальный вычислительный кластер

Экспериментальная часть

Генерацию трафика и запись лога между узлами будет осуществлять с помощью утилиты iperf. Размер генерируемых пакета по умолчанию 1500 байта. С каждого узла идет по 4 потока на порты узла Master: 80, 443, 145 и 137. Так же трафик генерируется между узлами Slave в один поток на порт 137.

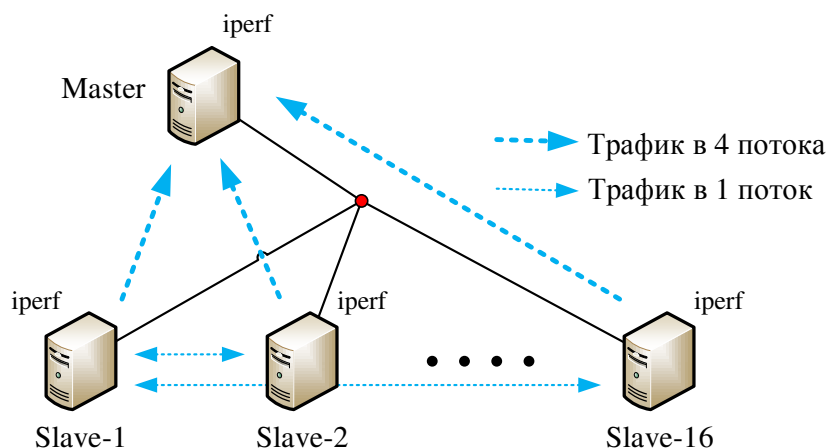


Рисунок 5 – Направление потока данных

Для автоматизации процесса запуска генерации трафика, был написан bash – скрипт.

Таблица 1. Результаты эксперимента

№ Узла	С протоколом OpenFlow		Без протокола OpenFlow	
	ΣОбъём данных(Мб)	Ср.скорость (Мб/с)	ΣОбъём данных(Мб)	Ср.скорость (Мб/с)
1.	58,12	5,13	1241	34,18
2.	561,48	35,94	1491,5	31,07
3.	57,45	3,01	2349,8	39,25
4.	838,78	33,38	2741,6	38,74
5.	1648,47	40,51	3036,3	36,19
6.	727,39	17,86	1920,65	20,01
7.	993,89	27,67	3254,88	33,13
8.	934,48	26,98	2938,14	32,68
9.	648,42	27,18	2900,35	33,62

10.	327,21	12,27	1851,75	21,11
11.	279,95	12,91	1863,03	21,88
12.	598,61	31,30	2909,21	34,73
13.	343,39	41,12	2942,78	32,35
14.	307,94	42,84	2898,92	32,28
15.	50,08	8,19	1853,84	22,09
16.	187,77	31,30	1915,66	21,88
Сумма/Ср.знач.	8375,66	24,85	38109,41	30,32

Заключение

Из полученных данных видим, что при использовании протокола OpenFlow объём сгенерированного трафика, а также средняя скорость передачи меньше чем без использования протокола. Утилита iperf генерирует трафик, тем самым максимально заполняет ширину полосы пропускания канала. С протоколом OpenFlow, утилита iperf создала много пустых отчет-файлов. После их обработки объём трафика получился 8,5ГБ, а без протокола 38ГБ. Это связано с производительностью коммутатора и контроллера OpenFlow. [6, 7] Другими словами, OpenFlow только обеспечивает базовые функции управления аппаратным обеспечением – коммутаторами, но не программную начинку SDN. И хотя многие ведущие производители сетевого оборудования, и в первую очередь – производители коммутаторов, заявляют о своей поддержке OpenFlow, программный интерфейс от контроллера к приложениям либо вообще недоступен, либо является собственной разработкой производителя. Это, безусловно, не позволяет архитектуре SDN полностью раскрыть свой инновационный потенциал, поскольку делает невозможной разработку «сетевых мозгов», независимых от производителя контроллера.

Литература

1. Википедия Свободная энциклопедия: https://ru.wikipedia.org/wiki/Облачные_вычисления
2. Mazikglobal: <http://www.mazikglobal.com/blog/cloud-computing-stack-saas-paas-iaas/>
3. Журнал сетевых решений LAN: <http://www.osp.ru/lan/2012/12/13033012/>
4. OpenNebula официальный сайт: <http://opennebula.org>
5. Википедия Свободная энциклопедия: http://ru.wikipedia.org/wiki/Java_Development_Kit
6. Малахов С.В., Тарасов В.Н. Экспериментальные исследования производительности сегмента программно-конфигурируемых // Интеллект. Инновации. Инвестиции 2013. №2. С81-85
7. Малахов С.В., Тарасов В.Н. Исследование производительности контроллера в программно-конфигурируемых сетях // Инфокоммуникационные технологии 2014. № 3. С64-67.