

# АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ В СИСТЕМАХ ОБРАБОТКИ И ХРАНЕНИЯ ДАННЫХ

В.Н. Тарасов, С.В. Малахов

Поволжский государственный университет телекоммуникаций и информатики

Данная статья посвящена виртуализации вычислительных кластеров, построенных на программно-конфигурируемых сетях (ПКС). Рассмотрен пример построения виртуального вычислительного кластера и анализ производительности ПКС. Определена задержка, вносимая использованием протокола OpenFlow.

## Введение

Сетевые архитектуры, в которых уровни управления (например, определение маршрута и качества передачи данных) отделены от уровней данных (т.е. пересылка пакета с одного порта на другой организована согласно таблице коммутации) становятся все больше популярными. Среди главных аргументов этого является то, что такая архитектура обеспечивает более структурированную программную среду для развития всей сети, потенциально упрощая уровень передачи данных. Управление передачей данных происходит с помощью специального программного обеспечения, так называемого контроллера, работающего на отдельном сервере, при этом контроллер вносит задержку обработки потоков. В связи с этим становится актуальной задача исследования производительности контроллера [1, 2].

## Программно-конфигурируемые сети в центрах обработки данных

Задача управления сетевыми ресурсами в настоящем времени возникает в любой компьютерной сети, а в центрах обработки данных является приоритетной. Ресурсы в сети обычно представляются как абстрактная единица, которая каким-либо образом потребляется запущенными задачами [3]. Но данный подход не учитывает специфику взаимодействия ресурсов друг с другом и с сетевой инфраструктурой. Сетевые ресурсы – ресурсы доступные через компьютерную сеть, следовательно, все, что может использоваться удаленно. Самые часто использующиеся ресурсы в центрах обработки данных – вычислительные мощности (процессорное время), оперативная память, долговременная память (хранение данных), каналы связи. Каждый из этих ресурсов используется своим образом и может влиять на другие ресурсы. Кроме этого, есть косвенные ресурсы, такие как потребляемая мощность и накладные расходы, являющиеся побочным следствием использования основных ресурсов. При реализации крупных проектов по созданию универсальных ЦОД на первом этапе эффективность ЦОД зависит, главным образом, от производительности компонентов и правильных структурных решений. Однако, на этапе эксплуатации ЦОД эффективность может снижаться из-за повышения накладных расходов на переконфигурирование, изменение структурных схем, простоя оборудования при переналадке и т.д.

Данные проблемы может решить появление принципиально нового подхода, называемого программно-конфигурируемыми сетями (ПКС – Software Defined Networks). ПКС-подход обещает сделать все сети дешевле и проще в управлении. Влияние ПКС-подхода будет ощущаться в центрах обработки данных (дата-центрах), корпоративных сетях, WAN, сотовых сетях, а также и в домашних условиях. Так же подход виртуализации распространяется сегодня не только на сервера, но и на сети, и на системы хранения. Средства ПКС позволяют эффективнее и проще управлять облачными конфигурациями. Для корпоративного сектора это возможность управления и оптимизации сложной инфраструктурой, а для среднего и малого бизнеса — инструмент эффективной работы с публичными облаками.

В основе ПКС сетей лежит представление о компьютерной сети, как сети, имеющей «плоскость данных», которая отвечает за пересылку пакетов на основе состояния в каждом коммутаторе, и «плоскости управления», которая отвечает за вычисление, «планирование» и управление пересылкой.

### Определение задержек

Необходимо произвести по 10 тестовых задач с поддержкой протокола OpenFlow и без, для оценки влияния протокола на производительность сети. Для исследования поведения пакетов было решено построить экспериментальный сегмент сети (Рисунок 1).

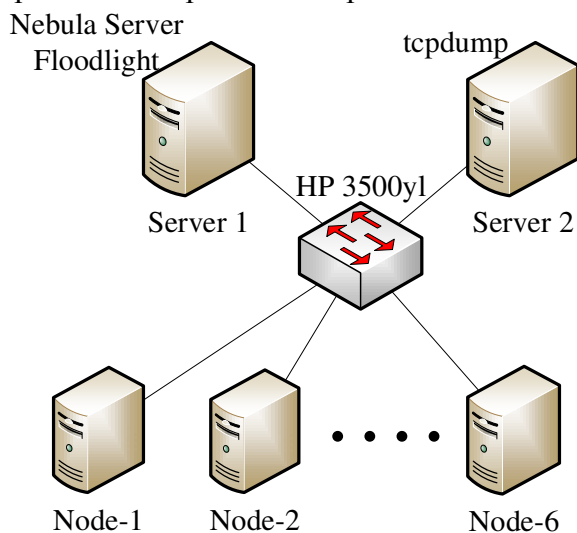


Рисунок 1 – Экспериментальный сегмент сети

На Server 1 установлен контроллер OpenNebula [4], а на сервера Node виртуальные вычислительные узлы кластера. В качестве OpenFlow контроллера (сетевой операционной системы) установлен Floodlight Open SDN Controller с приложениями по умолчанию.

На серверах, Node, будет развернут виртуальный вычислительный кластер, состоящий из 17-ти вычислительных узлов, с помощью OpenNebula (Рисунок 2).

Server 2 предназначен для захвата трафика с помощью tcpdump, подключенный к зеркалированному порту коммутатора.

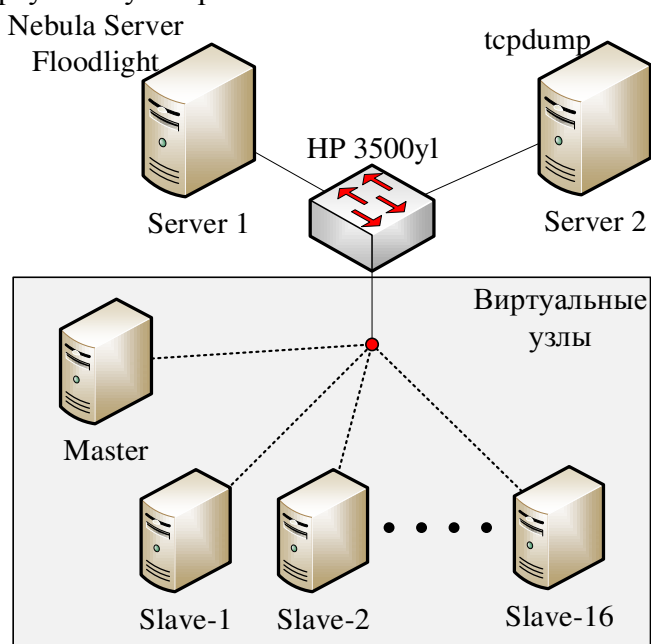


Рисунок 2 – Виртуальный вычислительный кластер

Под виртуальные вычислительные узлы выделены следующие ресурсы: Master – 1 CPU, ОЗУ 8ГБ и Slave – 1 CPU, ОЗУ 6ГБ.

### Генерация трафика

Трафик генерируется с помощью утилиты `iperf`, установленной на узлах кластера (Рисунок 3). Размер, генерируемых пакета по умолчанию 1500 байта. Обмен данными между машинами, по умолчанию 10 сек. На каждом из узлов запущена утилита `iperf` с параметром клиента, которая генерирует трафик в направлении узла Master. На Master утилита запущена с параметром сервера. 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. Тогда общее количество потоков за один тест составляет 304, а за эксперимент 3040.

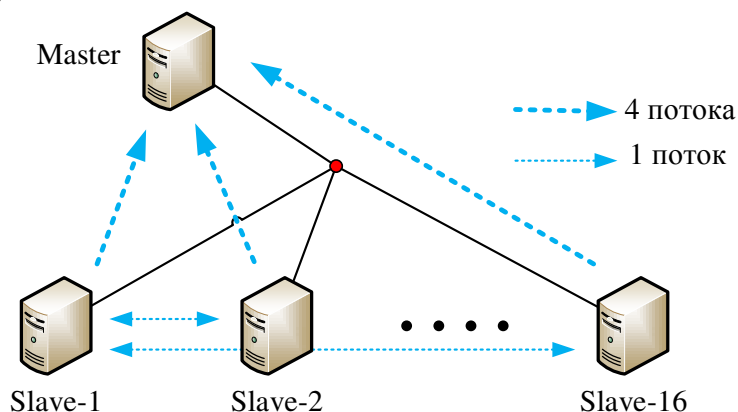


Рисунок 3 – Направление потока данных

Трафик до Server 1 зеркалируется на порт коммутатора (`#monitor all both mirror 1`), и захватывается `tcpdump` на сервере Server 2.

### Результаты выполнения эксперимента

Для анализа результатов понадобится обработать файлы логов выполнения эксперимента. На Server 2 файлы логов создаются утилитой `tcpdump`. За две части эксперимента сформировалось 20 файлов. Для обработки файлов потребуется программное обеспечение MS Excel.

1. Данные полученные без поддержки протокола OpenFlow. За 53,43 с. было зафиксировано в среднем 643813 пакетов. Среднее время задержки составляет 18,8 мс.

2. Данные полученные при поддержке протокола OpenFlow. За 31,18 с. было зафиксировано 127670 пакетов. Среднее время задержки составляет 3,6 мс.

Если размер пакета 1500 байт, то объем информации в направлении Server 1 будет 183МБ с поддержкой OpenFlow и 921МБ без.

Для сравнения задержки между двумя частями эксперимента необходимо рассмотреть поток данных на один порт с равным количеством пакетов. Максимальное количество обработанных пакетов на 80 порту, с поддержкой протокола OpenFlow, в среднем 3500. А без поддержки протокола, таких пакетов, обработанных сервером, составило в среднем 20000. Возьмем равное количество пакетов для каждого из 10 тестов, как с OpenFlow, так и без. Будем учитывать количество первых пакетов на 80-ый порт, остальные отбросим.

### Выводы

Малое количество пакетов с OpenFlow означает что, после обучения коммутатор перестает отправлять пакеты к контроллеру.

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 объем трафика получился 8,5ГБ, а без протокола 38ГБ. Это связано с производительностью коммутатора и контроллера OpenFlow.

Задержка при равном количестве пакетов на 80-ый порт, без поддержки протокола OpenFlow, равна 3,12 мс. Что меньше на 0,48 мс, чем с поддержкой протокола. Эта разница является задержкой, вносимая использованием протокола OpenFlow.

Из полученных результатов видно, что вычислительный кластер с протоколом OpenFlow не справился с предложенной нагрузкой.

#### **Литература**

1. Малахов С.В., Тарасов В.Н. Экспериментальные исследования производительности сегмента программно-конфигурируемых // Интеллект. Инновации. Инвестиции 2013. №2. С81-85.
2. Малахов С.В., Тарасов В.Н. Исследование производительности контроллера в программно-конфигурируемых сетях // Инфокоммуникационные технологии 2014. № 3. С64-67.
3. Орин Т, Поличелли Д. Администрирование корпоративных сетей на основе Windows Server 2008 — Москва: Русская Редакция, 2009. — Т. 1, С. 528.
4. OpenNebula официальный сайт: <http://opennebula.org>.