УДК 004.7, 004.732

ИССЛЕДОВАНИЕ СЕТИ ХРАНЕНИЯ ДАННЫХ, ПОСТРОЕННОЙ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ OPENFLOW

Садов О.Л.[1], Власов Д.В.[1], Грудинин В.А.[1], Каирканов А.Б.[1], Сомс Л.Н.[1], Титов В.Б.[1], Хоружников С.Э.[1], Чугреев Д.А.[1], Шевель А.Е.[1], Шкребец А.Е.[1]

[1]*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики», Санкт-Петербург, Россия (197101, г. Санкт-Петербург, Кронверкский проспект, д. 49), e-mail: xse@vuztc.ru*

В работе представлено: исследование возможностей построения и изучение эксплуатационных характеристик сегментов программно-конфигурируемых сетей (ПКС). Оценивается применимость ПКС для центров обработки данных (ЦОД) и распределенных систем хранения данных (СХД). Описываются эксперименты по нагрузочному тестированию сетей хранения данных с использованием контроллера NOX и различных программных и аппаратных OpenFlow-коммутаторов. Производится проверка на возможность обработки массовых запросов, измеряется задержка отклика, проводятся эксперименты по моделированию работы ЦОД из 64 коммутаторов и 100 тысяч хостов. Приводятся сравнительные результаты эффективности применяемых специфических механизмов обеспечения методов QoS. Дается описание основных проблем существующих реализаций компонентов OpenFlow ПКС, выявленных в ходе исполнения работ, даются рекомендации по возможным путям их решения.

Ключевые слова: программно-конфигурируемая сеть (ПКС), сеть хранения данных, система хранения данных (СХД), центр обработки данных (ЦОД), OpenFlow.

EVALUATION OF THE STORAGE AREA NETWORK DEVELOPED WITH SOFTWARE DEFINED NETWORKS OPENFLOW APPROACH

Sadov O.L.[1], Vlasov D.V.[1], Grudinin V.A.[1], Kairkanov A.B.[1], Soms L.N.[1], Titov V.B.[1], Khoruzhnikov S.E.[1], Chugreev D.A.[1], Shevel A.E.[1], Shkrebets A.E.[1]

[1]*National Research University of Information Technologies, Mechanics and Optics, Saint-Petersburg, Russia (197101, Saint-Petersburg, Kronverkskiy pr., 49), e-mail: xse@vuztc.ru*

Here it is described a study of possibilities to build up and test the performance of the software defined network (SDN) segments. The applicability of the SDN for data centers and the distributed storage systems. We describe experiments on stress testing of the storage area networks with the use of NOX controller and various software and hardware OpenFlow switches. A check on the possibility of processing bulk client requests, measured response delay, carried out experiments to simulate the operation of data center with 64 switches and 100 thousands hosts. The comparative results of the effectiveness of specific mechanisms implementing QoS are presented. The main problems of existing implementations of OpenFlow SDN components identified during the work and recommendations on possible ways to address them are described.

Keywords: Software-Defined Networks (SDN), Storage Area Network (SAN), Storages, Data Center, OpenFLow.

Введение

Традиционная архитектура компьютерных сетей (КС) уже сейчас перестает справляться с постоянным увеличением объемов и разнообразия передаваемых данных. Одна из технологий, способных вывести КС из кризиса, — программно-конфигурируемые сети (ПКС, Software Defined Networks, SDN) [4]. Концепция ПКС заявляет фундаментально иной подход к построению сетей. Основным принципом является снятие функционала по управлению сетями с устройств, главная цель которых — передача данных; управление же при этом должно осуществляться программными контроллерами на отдельных серверах.

Эффект от применения данной концепции может быть особенно ощутимым для сложных сетей центров обработки данных (ЦОД), где высокая степень виртуализации требует постоянной реконфигурации сети и тщательного управления потоками данных. Следует перечислить основные мотивы применения ПКС в ЦОД:

– снижение стоимости сетевого оборудования и его обслуживания за счет унификации и упрощения;

– повышение степени автоматизации управления сетью;

– повышение уровня защищенности за счет централизованного управления;

– возможность создавать высокоуровневые программные компоненты, реализующие любые алгоритмы управления ресурсами и потоками данных.

Наиболее известная и динамично развивающаяся реализация принципов ПКС – протокол OpenFlow [5]. Основной предоставляемый этой спецификацией функционал – управление обработкой потоков данных: коммутаторы под управлением OpenFlow анализируют входящий трафик на предмет соответствия записям в таблицах потоков (flow tables), где указаны действия, которые требуется осуществить с этими потоками. Добавление и модификация записей в таблице потоков инициируются удаленно расположенным контроллером, который включает в себя как стандартные модули, управляющие коммуникацией с сетевым оборудованием по OpenFlow, так и пользовательские, реализующие саму логику управления потоками.

Постановка задачи

Были поставлены следующие задачи исследований:

– анализ и оценка применимости ПКС для ЦОД и распределенных систем хранения данных (СХД), исследование принципов построения таких сетей, определение возможных проблем при их создании;

– реализация прототипов программных систем, выполняющих управление потоками данных и сетевыми ресурсами, а также обеспечение качества обслуживания (Quality of Service, QoS) для ЦОД и распределенных СХД.

Состав экспериментального стенда

Аппаратно-программный комплекс, на котором проводились исследования, включает в себя следующие основные компоненты:

– аппаратные коммутаторы, поддерживающие спецификацию OpenFlow — Pica8 P-3920 под управлением OpenVSwitch (OVS) [6] и HP 3500-24G-PoE yl;

– контроллеры NOX [6];

– системы эмуляции сетей под управлением OpenFlow -- Mininet [8];

– системы хранения данных HP P4300 G2 7.2TB (с передачей данных по протоколу iSCSI).

Специализированный модуль контроллера

Для обеспечения возможности управления сетевыми ресурсами и потоками данных, а также методами обеспечения QoS на базе модуля NOX switch был разработан специализированный модуль switchqos.

Данный модуль обеспечивает функциональность вычисления маршрутов прохождения пакетов через набор коммутаторов OpenFlow и генерации таблиц потоков (flow table) для каждого из этих коммутаторов. При изменении топологии соединений и прерывании потока данных по истечении периода ожидания производится изменение маршрута, и в таблицы потоков коммутаторов загружаются соответствующие изменения. Управление, осуществляемое модулем, характеризуется в главной степени набором действий (actions), который определяет способ обработки соответствующего потоку трафика. Для осуществления возможности управления методами обеспечения QoS в числе действий задается привязка потока, определяемого по заданному набору TCP/UDP портов, к соответствующим уровням обслуживания QoS. Методы обеспечения QoS в OpenFlow-коммутаторах могут быть реализованы либо через механизм очередей (queue) OpenFlow (механизм 1), либо через традиционные механизмы IP ToS и VLAN PCP (механизм 2). Соответственно, разрабатываемый модуль должен иметь возможность формировать два набора действий. Для механизма 1 это всего одно действие — постановка в очередь (enqueue) с двумя параметрами — номером интерфейса и номером очереди. Для механизма 2 предусматривается два действия: изменение поля ToS или PCP на соответствующее требуемому качеству обслуживания значение, а затем — вывод пакета на заданный интерфейс. Были разработаны две версии модуля switchqos для поддержки разных видов коммутаторов.

Для управления приоритизацией трафика СХД в сегменте ПКС использовались специализированные программные средства для задания политик QoS на аппаратных коммутаторах. Задача стандартизации интерфейса управления QoS для коммутаторов под управлением разных платформ, например OVS и HP, не может быть полноценно решена из-за высокой степени различия предоставляемых этими платформами средств. В этих условиях в качестве основного требования для разрабатываемых средств было выбрано упрощение удаленного задания наиболее важных характеристик QoS — пропускных способностей для очередей и их приоритетов. Прототипы программ для управления QoS на удаленно расположенных коммутаторах HP 3500 и Pica8 под управлением OVS размещены в репозитории [3]. Они могут быть легко расширены в случае необходимости удаленного управления иными наборами характеристик.

Обработка массового потока запросов

Разработанный модуль switchqos в сравнении с исходным switch содержит ряд улучшений, направленных на увеличение скорости прохождения данных через управляемые коммутаторы.

Включена модификация генерируемых по умолчанию API NOX схем соответствия (match). Необходимость этого обусловлена отсутствием на используемых коммутаторах аппаратной поддержки схем, включающих в себя все поля пакета, в том числе MAC-адресов источника, получателя и VLAN PCP. Это изменение позволило генерировать потоки, обрабатываемые на аппаратном уровне всеми используемыми коммутаторами, устранив тем самым низкое ограничение на скорость программной обработки пакетов (10000 в секунду).

Значение времени простоя (idle_timeout), являющееся параметром генерируемых потоков, также изменено. Эксперименты показали, что коммутатор HP 3500 при работе с аппаратно обрабатываемыми потоками недостаточно часто обновляет статистику по количеству привязанных к потоку прошедших пакетов. Как правило, в случае равенства времени простоя 5 секундам (задаваемое модулем switch значение), коммутатор на основе этой статистики принимает ошибочное решение об отбрасывании потока. В модуле switchqos время простоя равно 20 секундам, что стабилизирует соответствующую потоку запись в таблице коммутатора в случае непрерывности течения привязываемого к нему трафика и исключает задержки на регенерацию потоков. Следует отметить, что слишком большие значения времени простоя также могут нанести ущерб производительности из-за роста таблиц потоков.

Тем самым на тестовом стенде удалось значительно повысить производительность системы и преодолеть порог 100 000 запросов к СХД в секунду через коммутатор под управлением OpenFlow. В программе, осуществляющей тестирование (rd_test), используется iSCSI-команда «TEST UNIT READY», отправляемая в многопоточном режиме на СХД через системный вызов SG_IO ioctl.

Измерялось количество выполненных запросов за указанный период времени, ниже приведен пример вызова.

```
# /home/test/rd_test/rd_test /dev/sdb 2 100
…
Result: 130124 requests/sec (260248/2)
```

Задержка отклика при обращении к СХД

Для проверки задержки отклика при обращении к СХД использовались операции чтения с хранилища данных. Для исключения влияния механизмов буферизации на ход измерений для чтения данных вместо стандартного read использовался механизм SG_IO ioctl.

В тестовой программе измерялось среднее время задержки (Latency) при выполнении запросов к СХД с и среднеквадратичное отклонение этой задержки (Jitter). При измерении на 1000 пакетов и размерах блоков данных в 512 и 1024 были получены следующие результаты (средняя задержка и отклонение измерялись в секундах):

```
# ./rtt_iscsi_read /dev/raw/raw1 1000 512 1024
Size=512 Packets=1000 Latency=0.000844 Jitter=0.000084
Size=1024 Packets=1000 Latency=0.000860 Jitter=0.000104
```

Моделирование работы ЦОД

Моделирование ЦОД выполнялось при помощи отправки ICMP запросов на СХД с различных MAC-адресов. Использовалась сеть из СХД, двух аппаратных OpenFlow-коммутаторов HP, виртуальной машины под управлением ОС НауЛинукс 6.3 (головной хост) с головным контроллером NOX и десяти тестовых узлов, созданных на базе виртуальных машин под управлением ОС Ubuntu Linux 12.04 с системой эмуляции OpenFlow сети Mininet. На каждом из узлов запускалось по 6 виртуальных хостов, 7 программных коммутаторов OpenVSwitch и локальный контроллер NOX в соответствии с рисунком 1.

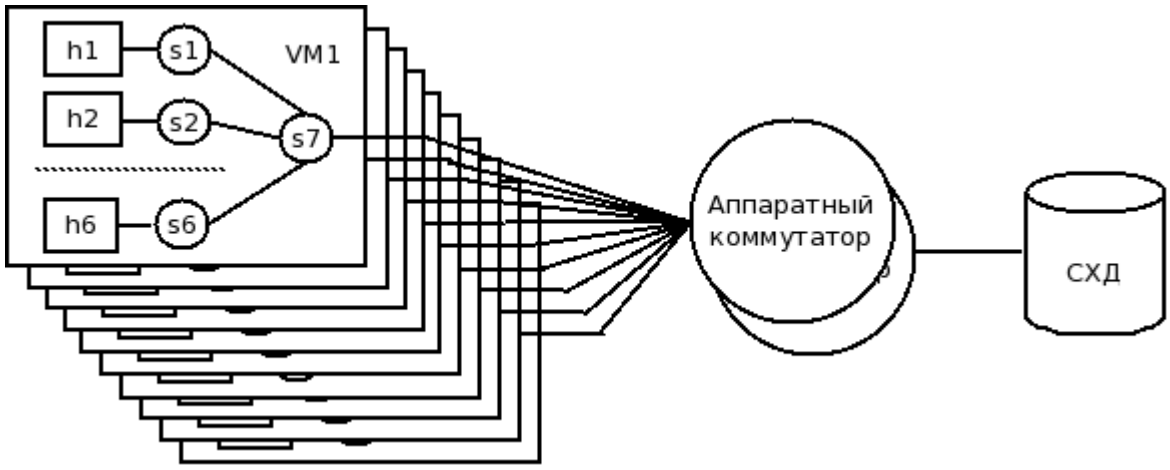

Рисунок 1 - Схема моделирования массовых запросов к СХД в инфраструктуре ЦОД

Тестовая программа, запускаемая на головном хосте, посылала сообщения на тестовые узлы, инициирующие запуск локальных тестовых программ, оформленных как xinetd-сервисы. Локальные тестовые программы на каждом из виртуальных хостов выполняли операции ping на СХД с перебором MAC-адресов в заданном диапазоне. Запросы направлялись к СХД через аппаратные коммутаторы, управляемые контроллером NOX, запущенным в многопоточном режиме (10 потоков) на головном хосте. Этот экземпляр контроллера протоколировал количество MAC-адресов, с которых посылались запросы, и распределение запросов по потокам в специальных файлах. Исполнение тестовой программы завершалось по достижении 100 000 хостов.

Выводы

В ходе выполнения описанного ряда экспериментов с разработанными прототипами приложений контроллера продемонстрированы низкие задержки на передачу данных и работоспособность системы при большом количестве хостов. Таким образом, можно сделать вывод, что решения на базе ПКС/OpenFlow для ЦОД и распределенных СХД могут быть достаточно эффективными и отвечать высоким требованиям к масштабируемости и быстродействию.

Проблемы применения этих технологий на сегодняшний день связаны в основном с недостаточной их распространенностью: следует отметить слабую поддержку OpenFlow аппаратными коммутаторами, отсутствие единых интерфейсов управления QoS, ошибки в работе контроллера.

В программных репозиториях [2] и [3] размещены разработанные программные модули и тесты. Приложения контроллера также оформлены в виде бинарных пакетов и пакетов с исходным кодом для дистрибутива ОС Linux НауЛинукс [1].

Статья подготовлена на материале исследований, которые проводились при финансовой поддержке Министерства образования и науки Российской Федерации в рамках государственного контракта № 14.514.11.4045 от 01 марта 2013 г.

Список литературы

1. Операционная система НауЛинукс [Электронный ресурс]. - Режим доступа: http://www.naulinux.ru/ (дата обращения: 29.03.2013).
2. Репозитории контроллера OpenFlow ИТМО [Электронный ресурс]. - Режим доступа: https://github.com/itmo-infocom/ (дата обращения: 13.06.2013).
3. Репозиторий программного коммутатора ИТМО [Электронный ресурс]. - Режим доступа: https://github.com/itmo-infocom/of12softswitch/commit/d376d8dfe9973d3b74d1b17e83ebd532c5b4fd14 (дата обращения: 26.03.2013).
4. Смелянский Р.Л. Программно-конфигурируемые сети // Открытые системы. СУБД. – 2012. - № 9.
5. Спецификации OpenFlow [Электронный ресурс]. - Режим доступа: https://www.opennetworking.org/sdn-resources/onf-specifications/openflow (дата обращения: 13.06.2013).
6. CPqD/nox12oflib [Электронный ресурс]. - Режим доступа: https://github.com/CPqD/nox12oflib (дата обращения: 28.03.2013).
7. CPqD/of12softswitch [Электронный ресурс]. - Режим доступа:

https://github.com/CPqD/of12softswitch (дата обращения: 28.03.2013).
8. Mininet [Электронный ресурс]. - Режим доступа: http://mininet.github.com/ (дата обращения: 28.03.2013).

Рецензенты:

Парфенов В.Г., д.т.н., профессор, декан факультета информационных технологий и программирования Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (НИУ ИТМО), г. Санкт-Петербург.

Горелик С.Л., д.т.н., профессор Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (НИУ ИТМО), г. Санкт-Петербург.