АРХИТЕКТУРА FULL-PROXY центра обработки данных

А.Серебряков, системный инженер компании F5 Networks

В период появления первых решений для балансировки нагрузки и доставки приложений существовала путаница в определении понятия и функций прокси-устройств и, в частности, самой архитектуры Full-Proxy. Она приобретает все большее значение в сегодняшнем процессе адаптации центра обработки данных с целью поддержки более мобильной, виртуализованной инфраструктуры для реализации сервисов.

Технология прямого прокси редко заслуживает нашего внимания, из-за того что функции эти часто обеспечиваются только на сетевом уровне и не затрагивают прикладной. С развитием технологий прикладного уровня прямое проксирование упоминается все чаще. Возникла необходимость в определении понятия и функций прокси-устройств и, в частности, прикладной Full-Proxy-архитектуры.

Платформа Full-Proxy

Различие систем прокси и Full-Proxy проявляется в способе обработки устройством сетевых соединений. Все прокси располагаются между двумя объектами – в эпоху Интернета это почти всегда клиент и сервер – и играют роль посредника в соединении. Все системы Full-Proxy являются прокси-устройствами, а вот обратное не всегда верно. Не все прокси являются Full-Proxy-системами, и эту особенность необходимо учитывать, принимая решения, которые могут повлиять на архитектуру центра обработки данных (ЦОД).

Система Full-Proxy поддерживает две отдельные таблицы сессий – одну на стороне клиента, другую на стороне сервера. Между ними создан изолирующий уровень, или своеобразный "зазор", позволяющий применять отдельные настройки для каждой из сторон при решении задач, актуальных только в определенном сегменте. Пользователи часто сталкиваются с большими задержками из-за узкой полосы пропускания. У сервера задержки, как правило, меньше, так как он использует высокоскоростную сеть ЦОД. Методы оптимизации и ускорения, которые применяются на стороне клиента и на стороне локальной сети, значительно отличаются друг от друга, поскольку проблемы, влияющие на производительность и доступность, в этих случаях совершенно разные.

Устройство Full-Proxy с раздельной обработкой соединения на каждой из сторон "зазора" способно справиться с этими проблемами. Прокси-система может формально быть и Full-Proxy, но обычно просто использует метод buffer-and-stitch для управления соединением и не может эффективно решить указанные проблемы. Типичный прокси осуществляет буферизацию соединения в начале TCP-сессии и иногда при передаче нескольких первых пакетов с данными приложения, а затем закрепляет (stitch) соединение в таблице сессии, пусть и с использованием данных прикладного уровня. В этом случае соединение будет представлять собой единый сквозной поток данных, и прокси придется выбирать, какие характеристики (клиентские или серверные) к нему применить, так как одновременно оптимизировать и те и другие не удастся.

Второе преимущество архитектуры Full-Proxy состоит в том, что она способна реализовывать больше возможностей обработки передаваемых данных с помощью различных функциональных

подсистем. Благодаря своему способу установки соединения Full-Proxy может направить на обработку той или иной функциональной подсистемой только те данные, которые ей необходимы. Каждая подсистема в свою очередь предпринимает специальные действия для анализа, обработки или модификации данных и отправляет их следующему компоненту. Это позволяет инспектировать SSL-соединения (Secure Sockets Layer – уровень защищенных сокетов), применять прикладные политики безопасности, предоставлять сервисы с заданным уровнем производительности отдельным клиентам или приложениям и одновременно обеспечивать наилучший уровень производительности за счет экономии ресурсов системы.

Такие возможности все более широко используются в архитектуре ЦОД для реализации уровня доставки приложений, на котором проблемы эксплуатационных рисков могут быть решены за счет принудительного применения различных политик. Фактически F5 создала архитектуру Full-Proxy-ЦОД, в которой уровень доставки приложений является посредником между клиентами и приложениями, осуществляя полное разделение их взаимодействия.

Архитектура Full-Proxy-ЦОД

Архитектура Full-Proxy-ЦОД создает коммутационный "зазор" между пользователем и приложениями, играя роль агрегации (и, наоборот, разделения) для сервисов. Поскольку сегодня все коммуникации сосредоточены в виртуализованных приложениях и сервисах на уровне доставки приложений, этот уровень служит стратегической точкой управления, на нем можно реализовать политики доставки для снижения эксплуатационных рисков (производительности, доступности, безопасности).

Преимущество архитектуры Full-Proxy-ЦОД заключается в изолировании конечных пользователей от неустойчивости нагрузки, свойственной высоковиртуализованным и динамичным облачным средам. Это позволяет создавать решения, предназначенные для преодоления ограничений технологии виртуализации, подобных ограничениям POD-архитектуры в средах VMware View. Традиционно технологии управления доступом оперируют именами хостов и IP-адресами для определения объектов доступа. В высоковиртуализованных или облачных средах это ограничение может грозить катастрофическими последствиями для производительности или даже работоспособности сервиса. Реализация управления доступом

на уровне доставки приложений – на устройстве Full-Proxy – позволяет устранить эту неустойчивость путем виртуализации своих ресурсов. При этом контроллер доставки приложения берет на себя такие детали, как знание IP-адресов и VLAN (Virtual Local Area Network), а само решение для управления доступом определяет, разрешено ли данному пользователю или данному устройству получать доступ к определенному сервису.

По существу, F5 просто расширила концепцию Full-Proxy на всю архитектуру. Введение уровня доставки приложений делает архитектуру более адаптируемой, гибкой, способной реагировать на быстрые изменения, с которыми приходится сталкиваться современным ИТ-подразделениям.

Кроме того, этот уровень предоставляет эффективные средства противодействия современным атакам. Возможность изолировать приложения, сервисы и даже ресурсы инфраструктуры позволяет уровню доставки приложений повышать способность организации противостоять согласованным DDoS-атакам. Благодаря различиям в емкости подключения контроллера доставки приложений и значительной части инфраструктуры (и всех серверов) архитектура в целом имеет повышенный уровень устойчивости на случай чрезмерного количества подключений. Это гарантирует более высокую доступность и в совокупности с виртуальной инфраструктурой, способной масштабироваться по запросу, помогает поддерживать необходимые для бизнеса уровни производительности.

Архитектура Full-Proxy-ЦОД – неоценимый актив ИТ в решении проблем неустойчивости нагрузки как внутри ЦОД, так и за его пределами.

Программно определяемые сервисы приложений

Продолжая тему снижения эксплуатационных рисков при создании современных ИТ-сервисов, необходимо учитывать, что их создание и настройка (в том числе и программно управляемая) невозможны без наблюдения и балансировки нагрузки, и потому многие сервисы обеспечения производительности, безопасности и управления доступом стали неотъемлемой частью архитектуры современных ЦОД. Они призваны решить задачи, связанные с такими тенденциями, как Интернет вещей, мобильность, постоянные сетевые атаки и растущие ожидания пользователей.

Вопрос о том, что делать с сервисами 4–7-го уровня с учетом развития SDN-подхода (Software-defined Networking), обсуждался очень активно, на эту тему было сказано много, но никто не смог предложить действительно хорошее решение – такое, которое было бы интегрировано как с приложениями (оркестровка облачных вычислений и виртуализации), так и с решениями для оркестровки сети.

Компания F5 объединила свои интеллектуальные ресурсы и нашла решение, которое интегрировано и совместимо с сервисами SDN и с решениями для оркестровки вычислительных систем. Решение F5 Synthesis применяет принципы SDN к сетям сервисов приложений, абстрагирует сетевые ресурсы приложения для доставки сервисов приложений на уровне, находящемся между сетевым уровнем и уровнем самих серверов. Это решение позволяет реализовать программно определяемые сервисы приложений SDAS (Software Defined Application Services).

SDAS представляет собой результат доставки чрезвычайно гибких программных сервисов приложений из унифицированного высокопроизводительного набора сервисов (service fabric). Сервисы SDAS, интеллектуальная оркестровка которых выполняется решением BIG-IQ, могут быть разработаны и инициализированы для удовлетворения важнейших потребностей, возникающих в вихре современных ИТ-тенденций.

В основе SDAS лежит идея использовать преимущества ресурсов, объединенных в пулы физических, виртуальных и облачных платформ F5 в любых их сочетаниях. Конечные пользователи получают возможность выделять ресурсы сервисам, а не устройствам, и реализовывать преимущества системного подхода компании F5, основанного на платформе Full-Proxy.

Гибкие сервисы SDAS имеют программные интерфейсы не только на уровнях управления (iControl, REST API) и плоскости передачи данных (iRules, node.js, Groovy), но и на конфигурационном уровне (iCall и iApps), что позволяет изменять настройки на уровне сервисов в режиме реального времени на основе событий.

SDAS – это новый этап в эволюции доставки приложений. Поскольку подход к ЦОД становится все в большей степени программно определяемым, таким же должен быть и подход к компонентам ЦОД. К их числу, безусловно, должен относиться сервис доставки приложений, который стал критически важным для обеспечения надежности, безопасности и производительности растущего перечня приложений, предоставляемых сотрудникам и пользователям, и других составляющих современного Интернета. ■