ПРОКСИ-МОДЕЛИ В КОНТЕЙНЕРНОЙ СРЕДЕ

Лори МакВитти (Lori MacVittie), главный технический евангелист F5 Networks, США

Inline, side-arm, reverse, и forward. Этими терминами мы ранее описывали место прокси-серверов в архитектуре сети.

Сегодня контейнеры используют те же термины, но вводят и новые. Итак, я хотел бы воспользоваться предоставленной мне возможностью для импровизированного рассказа на мою излюбленную тему: о прокси-серверах.

Масштабируемость была и остается одной из основных движущих сил развития облачных технологий (если забыть про свои несбыточные мечты об уменьшении затрат). Масштабируемость бросала вызов быстродействию (а иногда и побеждала его) в различных опросах, проводившихся на протяжении последних пяти лет, как эффект номер один, к которому стремятся организации, развертывая свои приложения в облачных окружениях.

Частично это связано с тем, что в цифровой экономике (в которой мы теперь работаем), прикладные программы стали эквивалентом реальных табличек «закрыто/открыто» и символом помощи цифровым потребителям. Медленные, не отвечающие на запросы приложения влияют на клиентов так же, как погашенный в магазине свет или недостаточное количество продавцов.

Приложения должны быть доступными и реагировать на запросы. Масштабируемость — это технический ответ на стремление к достижению этой бизнес-цели. Облако дает не просто возможность масштабировать, но и возможность осуществлять масштабирование автоматически. Для этого нужен балансировщик нагрузок. Ведь именно таким образом мы масштабируем приложения: при помощи прокси, которые балансируют нагрузку, связанную с трафиком и запросами.

Failing to respond to a digital visitor is the same as hanging a "closed" sign on the shop. The smallest delay can incur huge losses, especially for organizations in the retail industry.

С точки зрения ожиданий по поводу масштабируемости, контейнеры от них ничем не отличаются. Контейнеры должны масштабироваться — причем, автоматически, и это значит, что нам нужны балансировщики нагрузки (прокси).

Если вы используете собственные возможности — это значит, что вы выполняете примитивное балансирование нагрузки на основе TCP/UDP. В общем, реализация прокси на основе

контейнеров не слишком хорошо справляются с HTTP или другими протоколами уровня приложений, и не предлагают никаких возможностей, выходящих за пределы старого доброго балансирования нагрузки (POLB). Зачастую этого вполне хватает: масштабирование контейнеров работает на основе горизонтального принципа клонирования: для масштабирования приложения нужно добавить еще одну его копию и направить на нее запросы. Возможности маршрутизации седьмого уровня (HTTP) обнаруживаются на входах систем (во входных контроллерах и шлюзах API) и используются не столько для маршрутизации, сколько для их масштабирования.контейнеров не слишком хорошо справляются с HTTP или другими протоколами уровня приложений, и не предлагают никаких возможностей, выходящих за пределы старого доброго балансирования нагрузки (POLB). Зачастую этого вполне хватает: масштабирование контейнеров работает на основе горизонтального принципа клонирования: для масштабирования приложения нужно добавить еще одну его копию и направить на нее запросы. Возможности маршрутизации седьмого уровня (HTTP) обнаруживаются на входах систем (во входных контроллерах и шлюзах API) и используются не столько для маршрутизации, сколько для их масштабирования.

В некоторых случаях этого, однако, недостаточно. Если вам хочется (или нужно) обеспечить масштабируемость, ориентированную на приложения, или возможность добавлять дополнительные службы, вам придется воспользоваться более надежными предложениями, обеспечивающими возможность программирования или необходимую масштабируемость, или и то, и другое сразу.

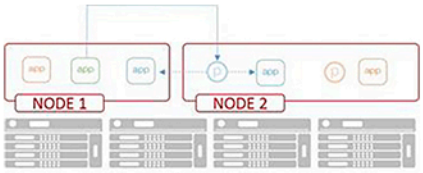
Для этого потребуется подключение прокси. То окружение управления контейнерами, в котором вы работаете, в значительной степени определяет модель развертывания прокси с точки зрения его вида: обратного или прямого. Для поддержания интереса скажу, что существует и третья модель — «боковая», и именно она является фундаментом масштабируемости, поддерживаемой новыми реализациями сервисных сетей.

Reverse proxy

Reverse proxy наиболее близок к традиционной модели, в которой виртуальный сервер получает все входящие запросы и распределяет их по пулу (ферме, кластеру) ресурсов.

При этом, на каждое «приложение» приходится по одному прокси. Любой клиент, желающий подключиться к приложению, соединяется вместо этого с прокси, который перенаправляет запрос на соответствующее приложение. Если зеленое приложение хочет пообщаться с синим, оно направляет запрос на синий прокси, который определяет, какой из двух экземпляров синего приложения должен ответить на запрос.

В этой модели прокси занимается только тем приложением, за которым он закреплен. Синий прокси ничего не знает об экземплярах, связанных с оранжевым прокси, и наоборот.



Per-App Reverse Proxy

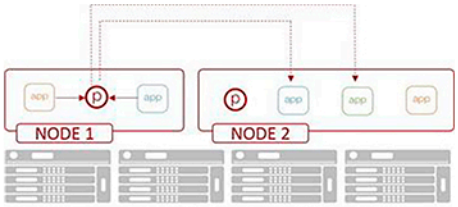
ПРОКСИ-МОДЕЛИ В КОНТЕЙНЕРНОЙ СРЕДЕ

Forward Proxy

Эта модель — ближе всего к модели традиционного межсетевого экрана, защищающего исходящие соединения.

В этой модели каждый контейнерный узел имеет связанный с ним прокси. Если клиент хочет подключиться к определенному приложению или конкретной службе, его вместо этого подключают к локальному прокси контейнерного узла, на котором работает клиент. Этот прокси затем выбирает соответствующий экземпляр приложения и перенаправляет туда запрос клиента.

И оранжевое, и синее приложения подключены к одному и тому же прокси, связанному с соответствующим узлом. Этот прокси определяет, какой экземпляр приложения, указанного в запросе, должен отправить ответ.



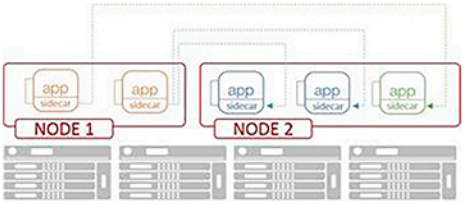
Per-Node Forward Proxy

В этой модели каждый прокси должен быть осведомлен о каждом приложении, чтобы быть в состоянии направлять запросы на соответствующий экземпляр приложения.

Sidecar Proxy

Этот режим работы известен также как «маршрутизатор служб сети». В этой модели каждый контейнер располагает своим собственным прокси.

Если клиент хочет подключиться к приложению, он соединяется с Sidecar Proxy, который выбирает соответствующий экземпляр приложения и перенаправляет туда запрос клиента. Пока все соответствует модели Forward Proxy.



Per-Pod (Container) Sidecar Proxy

У каждой из моделей есть свои преимущества и недостатки. Общей чертой всех моделей является зависимость от внешних данных (телеметрии и данных об изменениях в конфигурации), а также необходимости интеграции в экосистему. Некоторые модели диктуются выбранной вами средой, поэтому, прежде чем выбирать модель, необходимо тщательно оценить будущие потребности: в добавлении служб, безопасности, сложности сетевого взаимодействия.

ПРОКСИ-МОДЕЛИ В КОНТЕЙНЕРНОЙ СРЕДЕ

Мы находимся на ранней стадии развития контейнеров и их роста на корпоративном уровне. И по мере их появления в рабочей среде очень важно понимать потребности приложений, которые доставляются через контейнерное окружение, а также — как их прокси-модели разнятся в реализации.

Группа компаний БАКОТЕК – официальный дистрибьютор F5 Networks в Украине, Азербайджане, Республике Беларусь, Грузии, Армении и Молдове.
bakotech.com, f5@bakotech.com, +38 044 273 33 33.

F5 Networks, Inc.
401 Elliott Avenue West, Seattle,
WA 98119 888-882-4447 f5.com

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com

©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. 0113