



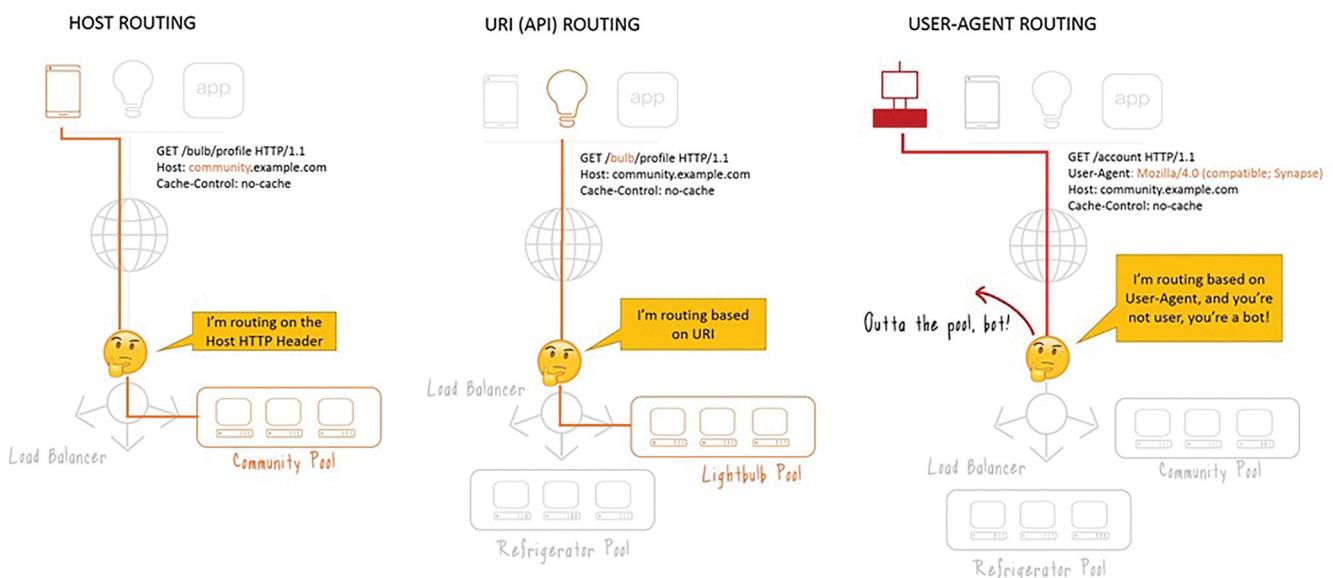
«БЫСТРЫЙ» API И МАРШРУТИЗАЦИЯ ПРИЛОЖЕНИЙ

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

Обычно маршрутизация приложений подразумевает использование полнофункционального прокси-сервера и возможность не только проверять, но и изменять заголовки и полезную нагрузку. При этом маршрутизация приложений играет все более важную роль в современной архитектуре компьютерных систем.

Кроме API существует много сценариев, единственной задачей которых является принятие решения о маршрутизации на основе HTTP-заголовка. Есть сценарий «Ты бот, дружище?», в котором необходимо попытаться [отсеять запросы, которыми управляют боты](#). Для этого необходимо проверять HTTP-заголовок «User-Agent». Другой сценарий – «Есть три хоста, какой из них интересует? (также известен как виртуальный хостинг)», – в котором нужно направлять входящий трафик на определенный хост, проверяя заголовок HTTP, скажем, «host». Еще один сценарий – «Какую версию этого API используем?», когда для определения версии API используется заголовок HTTP URI (Uniform Resource Identifier).

THREE MOST COMMON TYPES of APP ROUTING





В данных сценариях нужно просто проверить несколько заголовков HTTP 7-го уровня, а уже потом стандартные. Концепция входа контроллеров в ПО Kubernetes основана на том, что значения заголовка HTTP (URI или хоста) используются для обязательного направления запросов соответствующему сервису. Предполагается, что полезную нагрузку HTTP не нужно обрабатывать дополнительно *на прокси-сервере*. Возможно, придется вставить заголовок *X-Forwarded-For* или какой-либо другой настраиваемый заголовок HTTP, однако не нужно изменять существующие заголовки (например, переписывать URI) или проверять что-либо в реальной полезной нагрузке.

А если маршрутизации подлежит API (на универсальном идентификаторе ресурса URI или хоста) не имеет смысла замедлять процесс даже на доли секунды из-за использования уровня 7, ведь проверка уровня приложений добавляет небольшую задержку. HTTP – это текстовый протокол, значит он должен анализироваться и проверяться. Это занимает несколько процессорных тактов. Согласно закону Мура, подобные проверки выполняются быстро, однако если ограничиться известным набором строк, то проверка будет выполнена еще быстрее.

Идея такова: проверять лишь стандартные подзаголовки. Поскольку система знает, что ищет. В результате – высокая скорость маршрутизации приложений.

Высокая скорость маршрутизации приложений также является результатом по пакетной обработке запросов. Вот почему балансировка нагрузки на уровне 4 (TCP) очень быстра и масштабируема. Нет необходимости буферизовать пакеты, чтобы воссоздать HTTP-запрос целиком во время его проверки. Когда задействован уровень 7 (HTTP), например, при маршрутизации приложений – может потребоваться полнофункциональный прокси-сервер, способный буферизовать пакеты вплоть до получения полного HTTP-запроса или ответа.

Обладая этими возможностями, можно сделать много полезного. Но иногда есть необходимость маршрутизировать запросы на основе одного из нескольких стандартных заголовков HTTP, таких как **User-Agent**, **Host** или **URI**.

Предполагая, что данные запросы содержатся в одном пакете объемом менее 1500 байт, можно использовать [«быструю» версию маршрутизации приложений через прокси-сервер](#). Этот режим прокси-сервера сочетает скорость и масштабируемость балансировки нагрузки уровня 4 с интеллектуальными характеристиками уровня 7. Такое сочетание дает быстрый синтаксический анализ и не менее быстрый процесс принятия решения о маршрутизации. Прокси-сервер выполняет задачу маршрутизации за меньший промежуток времени и тратит на него меньше вычислительных ресурсов, что приводит к повышению производительности обработки запросов и ответов.



«БЫСТРЫЙ» API И МАРШРУТИЗАЦИЯ ПРИЛОЖЕНИЙ

Использование **маршрутизации приложений совместно с балансировкой нагрузки** очень полезно там, где нет большого общего пространства IP-адресов, которое можно расходовать как придется, а нужно использовать его разумно. Это решение позволяет использовать только одни IP-адреса для многих хостов. Оно может быть расширено с помощью дополнительного уровня маршрутизации приложений, например, контроллера входного трафика в кластере Kubernetes. Также можно использовать прокси-сервер, работающий с приложениями для дальнейшего очищения трафика, например, за счет выполнения некоторой сегментации данных или некоторых проверок безопасности приложений. Это решение также очень полезно при маршрутизации API. При этом имеется возможность открытой фильтрации версий или вызова сервисов на основе URI и перенаправления результатов на другой сервисный уровень, который выполняет более полную проверку или же лучше приспособлен для масштабирования в условиях конкретной вычислительной среды.

Сейчас практически любой прокси-сервер умеет выполнять маршрутизацию приложений, особенно на основе HTTP-заголовков. Но не каждый прокси-сервер может делать это быстро. Если для вас важна быстрота и масштабируемость, обязательно узнайте, может ли выбранный прокси-сервер работать в «быстром» режиме HTTP.



Группа компаний БАКОТЕК – официальный дистрибьютор F5 Networks в Украине, Азербайджане, Республике Беларусь, Грузии, Армении и Молдове.
<https://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