Изоляция рабочих нагрузок с использованием произвольного сегментирования

Colm MacCárthaigh

Изоляция рабочих нагрузок с использованием произвольного сегментирования

Сегодня Amazon Route 53 используется для размещения как крупнейших веб-сайтов для бизнеса, так и для наиболее популярных веб-сайтов в мире, однако начиналось все гораздо скромнее.

Размещение DNS: начало

Вскоре после того, как AWS начала предлагать услуги, клиенты AWS однозначно дали понять, что им нужно использовать сервисы Amazon Simple Storage Service (S3), Amazon Cloudfront и Elastic Load Balancing в «корне» своих доменов, т.е., для таких имен, как amazon.com, а не просто для www.amazon.com.

Казалось бы, ничего сложного. Однако из-за конструктивного решения, принятого при разработке протокола DNS в 1980-е годы, все несколько сложнее, чем кажется. Функция CNAME в DNS позволяет владельцу домена передать часть домена для размещения у другого провайдера, однако на корневом, или верхнем, уровне домена она не работает. Чтобы реализовать требования клиентов, нам пришлось действительно размещать их домены у себя. При размещении домена клиента мы можем возвращать любые наборы текущих IP-адресов для Amazon S3, Amazon CloudFront и Elastic Load Balancing. Эти услуги постоянно расширяются, а IP-адресов становится все больше, поэтому клиенты не могут зафиксировать их в своих доменных конфигурациях.

Таким образом, размещение DNS – задача масштабная. При проблемах с DNS весь бизнес может быть недоступен. И в то же время, определив задачу, мы принялись за ее решение так, как обычно делаем в Amazon, – срочно. Выделив группу инженеров, мы взялись за дело.

Защита от DDOS-атак

Спросите любого DNS-провайдера, чего он больше всего боится, и вам ответят – распределенных атак типа «отказ в обслуживании» (DDOS). Поскольку в основе DNS лежит протокол UDP, запросы DNS могут искажаться на большей части «дикого запада» Интернета. Поскольку DNS является критически важной инфраструктурой, такое сочетание делает ее привлекательной целью для кибервымогателей, «специалистов» по принудительному отключению сайтов, преследующих различные цели, и просто недалеких людей, которые не вполне понимают, что совершают серьезное преступление, влекущее за собой конкретные последствия. Но вне зависимости от причины каждый день на домены совершается несколько тысяч DDOS-атак.

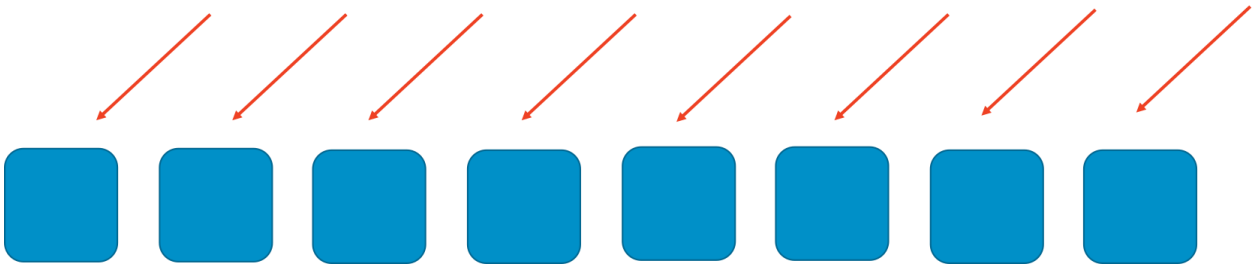
Одним из подходов к решению проблемы является существенное повышение серверных мощностей. Но мощность невозможно наращивать бесконечно, хотя определенный базовый уровень, конечно же, должен быть обеспечен. Каждый сервер, добавляемый провайдером, стоит несколько тысяч долларов, а киберпреступники могут увеличивать количество лже-клиентов практически бесплатно, используя зараженные бот-сети. Таким образом, стратегия увеличения количества серверов нежизнеспособна для провайдеров.

На момент создания Amazon Route 53 самой современной и совершенной защитой DNS считались специализированные сетевые устройства, которые умели быстро «фильтровать» трафик. Мы использовали целый ряд таких устройств в Amazon для собственных DNS-служб и в поисках решения задачи обратились к поставщикам оборудования. Выяснилось, что для закупки нужного количества таких устройств для каждого размещаемого на Route 53 домена потребовалось бы несколько десятков миллионов долларов и несколько месяцев для их поставки, установки и запуска в эксплуатацию. Но это не отвечало ни нашим срочным планам, ни нашим принципам разумной экономии, так что такой подход мы всерьез и не рассматривали. Нужно было найти такое решение, при котором защита появлялась бы только у атакуемых доменов. Старая пословица гласит: «голь на выдумку хитра». Нам нужно было «выдумать» DNS-сервис мирового класса со стопроцентной доступностью, который бы не потреблял много ресурсов. Результатом стало произвольное сегментирование.

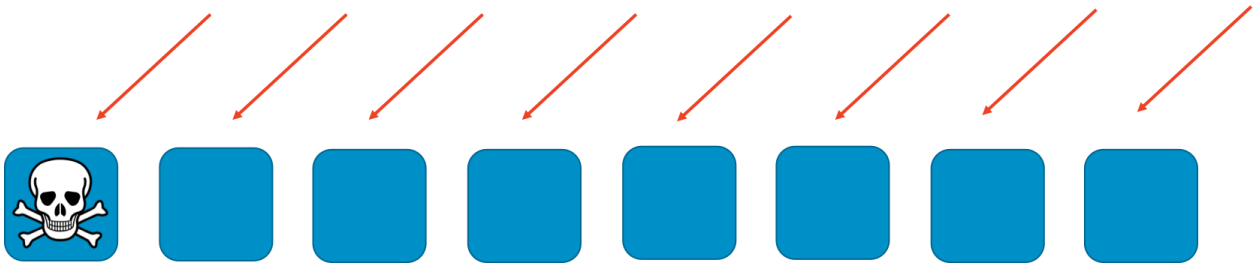
Что такое произвольное сегментирование?

Это простая, но мощная функция. Ее возможности даже шире, чем мы думали. Мы используем ее постоянно, и благодаря ей AWS обеспечивает эффективную и рентабельную работу многопользовательских сервисов при том, что ни один из клиентов не ощущает влияния других.

Чтобы понять, как работает произвольное сегментирование, сначала рассмотрим, как можно повысить масштабирование и отказоустойчивость системы при помощи обычного сегментирования. Представьте себе горизонтально масштабируемую систему или сервис, состоящую из восьми рабочих компонентов. Компонентов и их запросы показаны на следующем изображении. Компонентами могут быть серверы, очереди, базы данных – любая составная часть системы.



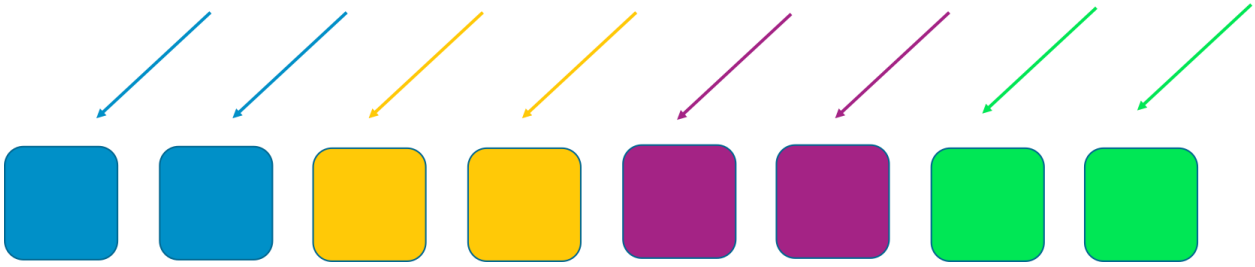
Без сегментирования парк компонентов выполняет всю работу. Каждый из компонентов может обработать любой запрос. Так обеспечивается эффективность и резервирование. При отказе одного компонента задачи распределяются между оставшимися семью, поэтому системе не требуется большой избыточной мощности. Проблема возникает тогда, когда определенный тип запроса или поток запросов, например DDOS-атака, могут привести к сбою. На следующих двух изображениях показан ход этой атаки.



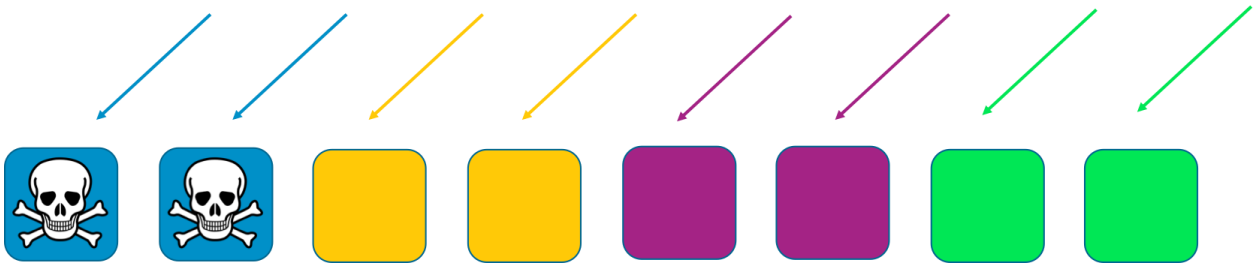
Из-за проблемы выходит из строя первый рабочий компонент, а затем она распространяется на другие компоненты, которые принимают на себя нагрузку. В результате очень быстро отказывают все компоненты, и сервис прекращает работу.

Отказ носит универсальный и всепоглощающий характер. Сервис полностью выходит из строя. Проблема затрагивает каждого клиента. Понятно, что такой подход неоптимален с точки зрения проектирования доступности.

Сегментирование позволяет улучшить ситуацию. Если мы поделим общий парк компонентов на 4 сегмента, то сможем, при меньшей производительности, снизить «масштаб катастрофы». На следующих двух изображениях мы видим, как сегментирование позволяет ограничить влияние DDOS-атаки.



В этом примере каждый сегмент состоит из двух рабочих компонентов. Ресурсы, например, клиентские домены, делятся между сегментами. Возможности резервирования сохраняются, однако поскольку в каждом сегменте только два рабочих компонента, то, чтобы справляться с отказами, системе требуется больше избыточной мощности. В обмен на это существенно снижается распространение проблемы.

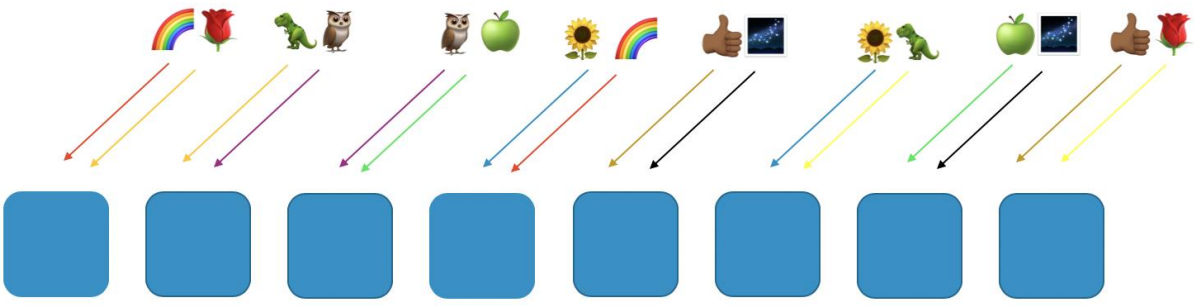


Ее охват сужается пропорционально количеству сегментов. При наличии четырех сегментов, если у клиента возникает проблема, то она, вероятно, затрагивает сегмент, где находится этот клиент, а также других клиентов в этом сегменте. В то же время, этот сегмент – только четверть сервиса. Деградация сервиса на 25 процентов значительно лучше, чем полное прекращение его работы. Произвольная сегментация позволяет выйти на новый уровень.

При произвольной сегментации мы создаем виртуальные сегменты с двумя рабочими компонентами в каждом и размещаем клиентов или ресурсы (или все то, что хотим изолировать) в одном из этих виртуальных сегментов.

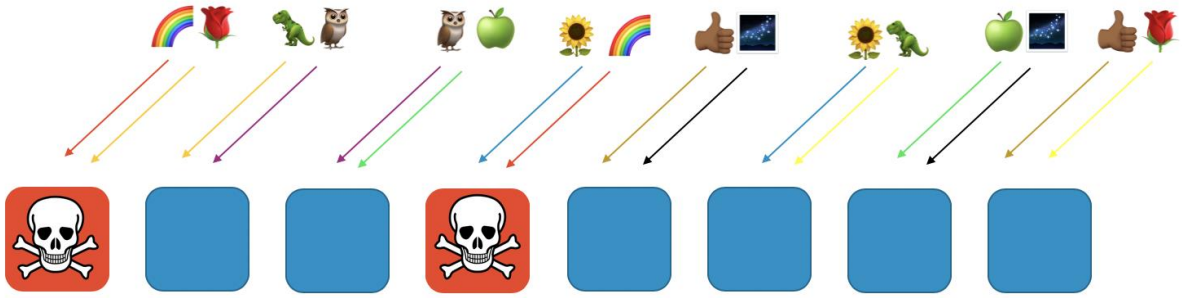
На следующем изображении показан пример схемы произвольного сегментирования с восемью рабочими компонентами и восемью клиентами, каждый из которых размещен в двух компонентах. Обычно клиентов значительно больше, чем рабочих компонентов, но в данном случае их небольшое количество делает ситуацию понятнее. Рассмотрим двух клиентов – «радугу» и «розу».

В нашем примере «радуга» размещается в первом и четвертом рабочих компонентах. Сочетание двух рабочих компонентов и является произвольным сегментированием для этого клиента. У других клиентов свои виртуальные сегменты с собственным сочетанием рабочих компонентов. Например, у «розы» – первый и восьмой.



Если у «радуги», которой выделены первый и четвертый рабочие компоненты, возникает проблема (например, вредоносный запрос или поток таких запросов), она затронет только этот виртуальный сегмент, но на другие она практически никак не повлияет. В действительности, она может повлиять максимум на еще один рабочий компонент

виртуального сегмента. Если сторона, выполняющая запросы, отказоустойчива и способна к обходу (например, выполняя повторные запросы), сервис оставшихся сегментов будет и далее работать бесперебойно для клиентов или ресурсов, как показано на следующем изображении.



Иными словами, хотя проблема может затрагивать все рабочие компоненты, которые обслуживают «радугу», она никак не затрагивает другие рабочие компоненты. Для клиентов это означает, что хотя клиенты «роза» и «ромашка» пользуются тем же рабочим компонентом, что и «радуга», проблема на них не отражается вообще никак. Как показано на следующем изображении, «роза» обслуживается восьмым рабочим компонентом, а «ромашка» – шестым.

При возникновении проблемы мы можем потерять четверть сервиса, но благодаря произвольной сегментации распределение клиентов или ресурсов таково, что негативное влияние проблемы существенно меньше. Восемь рабочих компонентов – это 28 уникальных комбинаций из двух таких компонентов, или 28 возможных произвольных сегментов. А если клиентов несколько сотен или даже больше и мы назначим каждому из клиентов произвольный сегмент, то масштаб проблемы сократится до 1/28. И это в 7 раз лучше, чем обычное сегментирование.

Приятно видеть, что показатели улучшаются по экспоненте с ростом количества рабочих компонентов и клиентов. Большую часть задач, связанных с масштабированием, при таком размахе решать сложнее, но произвольное сегментирование становится только эффективнее. В действительности, при достаточном количестве рабочих компонентов количество произвольных сегментов может превысить количество клиентов, и тогда можно изолировать каждого клиента.

Amazon Route 53 и произвольное сегментирование

Итак, как же все это помогает в работе Amazon Route 53? В Route 53 мы решили организовать общую емкость в 2048 виртуальных серверов имен. Это виртуальные сервера, так как они не соответствуют физическим серверам, на которых размещается Route 53. Мы можем перемещать их для управления мощностями. Далее мы назначаем каждому клиентскому домену произвольный сегмент из четырех серверов имен. Таким образом, количество произвольных сегментов увеличивается до потрясающей воображение цифры в 730 миллиардов. Их так много, что мы можем назначить уникальный произвольный сегмент каждому домену. И мы можем пойти еще дальше и сделать так, чтобы ни один клиентский домен не использовал бы более двух виртуальных серверов имен с любым другим клиентским доменом.

Результаты удивляют. Если на клиентский домен организуется DDOS-атака, то трафик у четырех серверов имен, связанных с ним, серьезно вырастет, но это никак не отразится на доменах других клиентов. И мы не успокаиваемся на том, что проблемы возникают только у одного клиента. Произвольная сегментация означает, что мы можем определять и изолировать клиента, ставшего целью конкретной атаки. Кроме того, мы разработали собственный уровень анализаторов трафика AWS Shield. В то же время, произвольная сегментация играет огромную роль в том, чтобы клиенты Route 53 не испытывали никаких проблем даже тогда, когда происходят описанные выше события.

Выводы

Мы реализовали произвольную сегментацию во многих других наших системах. Кроме того, мы постоянно совершенствуем ее, например, в виде рекурсивной произвольной сегментации, когда элементы сегментируются на целом ряде уровней, таким образом, изолируя клиента клиента. Произвольная сегментация обладает прекрасными возможностями адаптации. Это грамотный способ организации существующих ресурсов. Как правило, он не требует дополнительных затрат, поэтому способствует повышению рентабельности и экономичности.

Если вы тоже хотите попробовать произвольную сегментацию в деле, воспользуйтесь нашей открытой библиотекой Route 53 Infirma. В ней есть несколько вариантов реализации произвольной сегментации, которые можно применять для распределения или организации ресурсов.