Қазақстан Республикасы Тәуелсіздігінің 30 жылдығына арналған «Сейфуллин оқулары – 17: «Қазіргі аграрлық ғылым: цифрлық трансформация» атты халықаралық ғылыми – тәжірибелік конференцияға материалдар = Материалы международной научно – теоретической конференции «Сейфуллинские чтения – 17: «Современная аграрная наука: цифровая трансформация», посвященной 30 – летию Независимости Республики Казахстан.- 2021.- Т.1, Ч.4 - С.205-209

БАЛАНСИРОВКА НАГРУЗКИ СЕРВЕРА ПО МЕТОДУ SLB и HTTP

Шакирова Г.М.

В настоящее время разрабатываются сайты, способные обрабатывать миллионы одновременных запросов от пользователей. Основная задача этих сайтов - отвечать на запросы пользователей. Эти ответы состоят из изображений, текстов, видео, аудио и данных приложения. Для того, чтобы справиться с таким большим объемом данных, одним из простых решений является добавление количества серверов для обработки запросов пользователей. Это может привести к тому, что один сервер будет сильно загружен, а другой будет загружен меньше. Поэтому равномерная отправка запросов пользователей на все серверы является здесь сложной задачей. Для облегчения этой работы установлены балансировщики нагрузки. Первоочередной задачей балансировщика нагрузки является распределение сетевого трафика по различным серверам. Это повышает доступность веб-приложений и веб-сайтов для пользователей. Очень сложно работать с современными приложениями без балансировщиков нагрузки. Балансировка нагрузки серверов (SLB) распределяет большой трафик на несколько серверов с помощью программного или аппаратного обеспечения на базе сети.

В начале 1990-х годов была начата технология балансировки нагрузки для распределения сетевого трафика по сети. Контроллеры доставки приложений (ADC) обеспечивают унифицированный доступ и безопасность (ADC) приложений.

ADC делятся на три типа: аппаратные, виртуальные и программные устройства балансировки нагрузки. Услуги ADC включают балансировку нагрузки, разгрузку, кэширование, сжатие и безопасность. Таким образом, это обеспечивает более короткую доставку за меньшее время. Это достигается за счет балансировки нагрузки. Общая идея балансировки нагрузки серверов (Server Load Balancing - SLB) обсуждается в этой статье. Отправка запроса клиента на сервер является основной задачей SLB. Если балансировка нагрузки выполняется для другого геологического местоположения, то она считается Глобальной Балансировкой Загрузки Сервера (Global Server Load Balancing - GSLB). Такие серверы есть в

собственных центрах обработки данных компании или в публичных, или частных облаках. SLB препятствует сетевому трафику и перенаправляет поток сетевого трафика на серверы.

Несколько алгоритмов балансировки нагрузки обеспечивают сетевые услуги и доставку контента. Приоритет будет отдаваться конкретным запросам клиентов в сети. Повышение производительности и надежности гарантируется, когда SLB *(sticky load balancing)* распределяет клиентский трафик по серверам [1]. Алгоритм маршрутизации, основанный на трафике, не только оптимизирует сетевой ресурс на текущее время, но и на будущие запросы. Он также обеспечивает масштабируемость, быструю доставку и высокую доступность. Существует два метода выполнения SLB, таких как балансировка нагрузки на транспортном уровне и балансировка нагрузки на прикладном уровне. Балансировка нагрузки на уровне транспорта основана на DNS. Решения по балансировке нагрузки будут приниматься с помощью балансировки нагрузки на уровне приложений.

Преимущества SLB

Основной задачей SLB является равномерное распределение входящего трафика на несколько веб-серверов для повышения эффективности доставки приложений.

Когда уровень сетевого трафика приближается, достигает или превышает проектный максимум, сеть считается перегруженной. SLB работает не только как регулятор трафика, но и предоставляет такие преимущества, как прогнозный анализ, который идентифицирует перегруженность и дает решение для нее. Балансировка нагрузки является важным подходом к решению проблем перегруженности сети, возникающих из-за неэффективного распределения ресурсов. Маршрутизация может быть выполнена по подходящим маршрутам, а перегруженные маршруты можно избежать в соответствии с текущим спросом на трафик в сети.

В сетевой модели OSI балансировка нагрузки будет осуществляться на транспортном, сеансовом, презентационном и прикладном уровнях.

L4 уровень балансировки нагрузки, основанный на данных из сети и транспортного протокола.

L7 Уровень балансировки нагрузки дополняет переключение содержимого. Компетенции балансировки нагрузки L4 и L7 увеличиваются с помощью GSLB.

В настоящее время производительность балансировщиков нагрузки увеличена за счет размещения в центрах обработки данных собственных приложений, работающих в облаке.

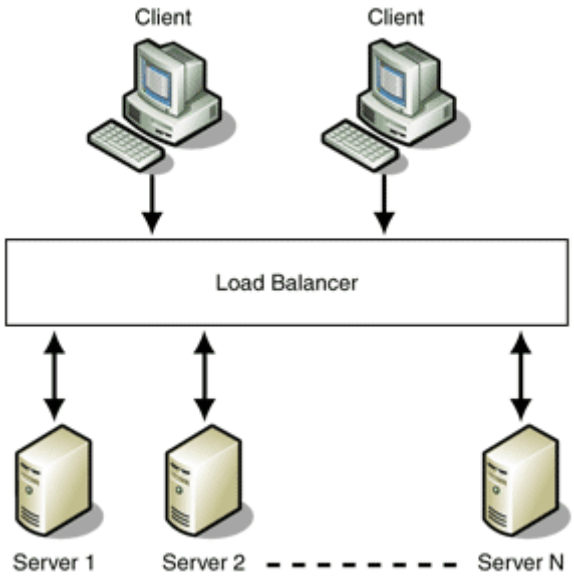

Рисунок 1. Балансировка нагрузки на сервер

Что делать, если какой-либо сервер вышел из строя? Наверное, оповестить об этом заинтересованных лиц и не направлять на сбойный сервер запросы до тех пор, пока заинтересованные лица (или специально обученная программа) не разберутся с возникшей проблемой. Не забывайте, что мощности оставшихся серверов должно быть достаточно для возросшей нагрузки. Если это будет не так, то может произойти коллапс всей системы.

В дальнейшем будем предполагать, что стратегия failover уже выбрана. Приступим к рассмотрению стратегий распределения нагрузки.

Разделим их на два семейства:

Cache-unaware. Это семейство стратегий не принимает во внимание возможное различие между данными, закэшированными локально на каждом сервере. Такие стратегии хорошо работают в двух случаях:

Если сервера вообще ничего не кэшируют локально (stateless servers). Например, если загрузка данных из удаленного ресурса по сети требует меньше процессорных ресурсов и времени, чем загрузка этих же данных из локального кэша. Такая ситуация возможна, если cache hit ratio для локального кэша стремится к нулю вследствие большого объема данных, которые требуется закэшировать. Также это возможно при частом обновлении данных, так что при очередном запросе они уже устаревают.

Если локальные кэши на всех серверах содержат одни и те же данные. В этом случае без разницы, куда будет направлен следующий запрос.

Для cache-unaware стратегий лучше всего подходит распределение нагрузки на наименее загруженный в данный момент сервер [2]. Эта стратегия позволяет добиться наименьшего времени ожидания в очереди запросов, в соответствии с результатами моделирования, описанными в моей предварительной статье.

Cache-aware. Это семейство стратегий направляет запросы таким образом, чтобы максимизировать количество cache hit’ов в локальных кэшах серверов.

Балансировка нагрузки HTTP-сервера

HTTP SLB - это архитектура запрос - ответ для потока HTTP-трафика. В сети каждый раз различные запросы размещаются на веб-сервере. Управлять этим трафиком будет очень сложно. Развертывание веб-сервера между сервером и клиентом облегчит эту проблему.

Преимущества Балансировки нагрузки HTTP

Оптимизация для конкретного приложения может быть выполнена эффективно. Реализация интуитивно понятного пользовательского интерфейса упрощает балансировку нагрузки. Реализован расширенный доступ к АЦП с 4-го по 7-й уровень.

Балансировка нагрузки TCP

Протокол управления передачей (TCP) используется для балансировки сетевого трафика. Связь между Интернет-протоколом (IP) и прикладной программой осуществляется по TCP-трафику. Для обеспечения правильной доставки каждому приложению присваивается уникальный номер порта. Балансировщик нагрузки приложения будет прекрасно работать на уровне TCP при условии, что уровень 4 модели OSI.

Для SLB необходимо определить, как распределять трафик через различные метрики. Обычно это может быть выполняется с помощью различных алгоритмов балансировки нагрузки, таких как *round robin, min-min, min-max* и т.д. Настройка процесса балансировки входящего трафика на неидентичные серверы зависит от вычислительной мощности серверов.

SLB предлагает следующие услуги:

1. Балансировка нагрузки в кластерах предлагается для распространения трафика и поддержания продолжительности сеанса через различные серверы.
2. Повышение эффективности обработки TCP и SSL.
3. Для эффективной балансировки нагрузки можно управлять множеством серверов.
4. Эффективная виртуальная балансировка нагрузки с полной функциональностью VMWare.

Как сгруппировать запросы одного и того же пользователя?

Обычно группировка осуществляется либо по IP-адресу входящего запроса, либо по идентификатору пользователя (например, user_id, session_id, auth_token). Идентификатор пользователя может находиться в различных местах. Например, в cookies, в HTTP header’ax, в url’e, в query string’e, в теле POST-запроса.

Главное преимущество группировки по IP в том, что она может быть осуществлена с минимальными затратами ресурсов на сетевом (IP) или транспортном (TCP) уровнях. Более того, в некоторых ОС типа Linux, группировка входящих TCP-подключений по Source IP встроена в ядро.

Изучите, например, опцию —persistent в DNAT target из IPtables или опцию —hashmode=SourceIP в CLUSTERIP target там же. Это позволяет построить высокопроизводительный load balancer без применения дополнительного софта. Правда, такой load balancer не сможет автоматически перенаправлять запросы в обход вышедших из строя серверов.

Основная условие данного принципа — равномерно распределить запросы, сгруппированные по IP или идентификатору пользователя, между имеющимися серверами.

Рассмотрим некоторые алгоритмы, удовлетворяющие этому условию:

Таблица ассоциаций. Для каждой группы запросов выбираем сервер с наименьшим количеством ассоциированных групп запросов, а соответствующую ассоциацию между группой запросов и сервером записываем в специальную таблицу ассоциаций load balancer'а.

Преимущества:

Идеальное распределение групп запросов на имеющиеся сервера.

Минимальная потеря ассоциаций при удалении серверов (failover) — теряются только ассоциации с удаленным сервером.

Отсутствие потерь ассоциаций при добавлении серверов — новые группы запросов будут добавляться в ассоциацию к новому серверу до тех пор, пока количество ассоциаций нового сервера не сравняется с количеством ассоциаций остальных серверов.

Злоумышленники не могут определить сервер, на который будет направлена данная группа запросов.

Недостатки:

Размер таблицы ассоциаций должен контролироваться, чтобы она не заняла всю доступную память в load balancer'е.

Т.к. при ограниченном размере таблицы ассоциаций старые ассоциации удаляются, то происходит их безвозвратная потеря [3]. Это означает, что группа запросов из удаленной ассоциации может быть ассоциирована с произвольным сервером в будущем.

При наличии нескольких load balancer'ов таблица ассоциаций должна быть синхронизирована между ними. Иначе они будут направлять запросы из одной и той же группы на различного сервера.

Таблица ассоциаций может быть безвозвратна утеряна при выходе из строя load balancer'а. В этом случае cache hit ratio резко упадет до нуля и будет оставаться низким, пока не заполнится новая таблица ассоциаций.

Балансировка нагрузки сервера решает, на какой сервер должен быть направлен запрос клиента для достижения высокой функциональности при минимальных затратах времени и средств. Вычислительная разгрузка перенесет огромные вычисления с мобильных на серверы в облаке. Облачные вычисления и мобильные облачные вычисления предназначены для выполнения всех вычислений с меньшими усилиями и вычислений с помощью SLB. В этой статье кратко обсуждаются возможности, методы и метрики SLB и HTTP. SLB методы и решения включают в себя ассимилированное администрирование потока трафика и ускорение

приложений. При неравномерном распределении нагрузок по пути перераспределение этих нагрузок по пути необходимо для поддержания хорошей производительности системы. Качество обслуживания SLB обеспечивает беспрецедентную власть над управлением потоком трафика и значительно улучшает выполнение приложений.

Список литературы

1. E.R. Naganathan, S. Rajagopalan and P. Herbert Raj, “Traffic Flow Analysis Model based Routing Protocol for Multi-Protocol Label Switching Network”, Journal of Computer Science 7 (11): 1674-1678, 2011, ISSN 1549-3636, © 2011 Science Publications.
2. P. Herbert Raj , P. Ravi Kumar and P. Jelciana, “Load Balancing in Mobile Cloud Computing using Bin Packing’s First Fit Decreasing Method”, Springer Nature Switzerland AG 2019, S. Omar et al. (Eds.): CIIS 2018, CIIS 2018, AISC Vol 888, pp. 97– 106, ISBN: 978-3-030-
03302-6_9, 2019.
3. Айвалиотис, Дмитрий Администрирование сервера NGINX / Дмитрий Айвалиотис. - М.: ДМК Пресс, 2013. - 1275 с.