

АНАЛИЗ УРОВНЕЙ БАЛАНСИРОВКИ НАГРУЗКИ В РАСПРЕДЕЛЁННЫХ СЕРВЕРАХ

Д.А. Хлебест, Е.С. Омелюсик

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шаталова В.В. – канд.техн.наук, доцент

Процедура балансировки осуществляется при помощи целого комплекса алгоритмов и методов, соответствующим следующим уровням модели OSI:

- сетевому;
- транспортному;
- прикладному.

Рассмотрим эти уровни более подробно.

Балансировка на сетевом уровне предполагает решение следующей задачи: нужно сделать так, чтобы за один конкретный IP-адрес сервера отвечали разные физические машины. Такая балансировка может осуществляться с помощью множества разнообразных способов.

DNS-балансировка. На одно доменное имя выделяется несколько IP-адресов. Сервер, на который будет направлен клиентский запрос, обычно определяется с помощью алгоритма Round Robin.

Построение NLB-кластера. При использовании этого способа серверы объединяются в кластер, состоящий из входных и вычислительных узлов. Распределение нагрузки осуществляется при помощи специального алгоритма. Используется в решениях от компании Microsoft.

Балансировка по IP с использованием дополнительного маршрутизатора.

Балансировка по территориальному признаку осуществляется путём размещения одинаковых сервисов с одинаковыми адресами в территориально различных регионах Интернета (так работает технология Anycast DNS). Балансировка по территориальному признаку также используется во многих CDN.

Балансировка на транспортном уровне является самым простым: клиент обращается к балансировщику, тот перенаправляет запрос одному из серверов, который и будет его обрабатывать. Выбор сервера, на котором будет обрабатываться запрос, может осуществляться в соответствии с самыми разными алгоритмами: путём простого кругового перебора, путём выбора наименее загруженного сервера из пула и т.п.

Иногда балансировку на транспортном уровне сложно отличить от балансировки на сетевом уровне. Рассмотрим следующее правило для сетевого фильтра pf в BSD-системах: так, например, формально тут идет речь про балансировку трафика на конкретном порту TCP (пример для сетевого фильтра pf в BSD-системах):

```
web_servers = "{ 10.0.0.10, 10.0.0.11, 10.0.0.13 }"  
match in on $ext_if proto tcp to port 80 rdr-to $web_servers round-robin  
sticky-address
```

Речь в нём идет о балансировке трафика на конкретном порту TCP.

Рассмотрим теперь другой пример:

```
pass in on $int_if from $lan_net \  
  route-to { ($ext_if1 $ext_gw1), ($ext_if2 $ext_gw2) }\  
  round-robin
```

В этом правиле речь о балансировке исходящего трафика на сетевом уровне. В нём не указано ни конкретного порта, ни конкретного протокола.

Различие между уровнями балансировки можно объяснить следующим образом. К сетевому уровню относятся решения, которые не терминируют на себе пользовательские сессии. Они просто перенаправляют трафик и не работают в проксирующем режиме.

На сетевом уровне балансировщик просто решает, на какой сервер передавать пакеты. Сессию с клиентом осуществляет сервер.

На транспортном уровне общение с клиентом замыкается на балансировщике, который работает как прокси. Он взаимодействует с серверами от своего имени, передавая информацию о клиенте в дополнительных данных и заголовках. Таким образом работает, например, популярный программный балансировщик HAProxy.

При **балансировке на прикладном уровне** балансировщик работает в режиме «умного прокси». Он анализирует клиентские запросы и перенаправляет их на разные серверы в зависимости от

характера запрашиваемого контента. Так работает, например, веб-сервер Nginx, распределяя запросы между фронтендом и бэкендом. За балансировку в Nginx отвечает модуль Upstream.

В качестве ещё одного примера инструмента балансировки на прикладном уровне можно привести rproxy — промежуточный слой между клиентом и сервером СУБД PostgreSQL. С его помощью можно распределять запросы оп серверам баз данных в зависимости от их содержания: например, запросы на чтение будут передаваться на один сервер, а запросы на запись — на другой.

Список использованных источников:

1. *Балансировка нагрузки: основные алгоритмы и методы [Электронный ресурс]. – Режим доступа: <https://selectel.ru/blog/balansirovka-nagruzki-osnovnye-algoritmy-i-metody>. – Дата доступа: 01.04.2020.*
2. *S.Sharma, S.Singh, M.Sharma. Анализ производительности алгоритмов балансировки нагрузки: всемирная академия наук, инженерии и технологии, 2008.*