

ИУ7-54Б, 16_KOZ, Р_{унов}

1 Введение

В современном мире активно развиваются интернет-компании. По мере роста, им приходится иметь дело с постоянно возрастающими количеством пользователей и объемом обрабатываемых данных, и в какой-то момент возникает потребность в увеличении количества серверов, а затем – в эффективном распределении нагрузки между ними. Для этого используются так называемые **балансировщики нагрузки**. Балансировщик нагрузки выступает в роли «регулирующего», стоящего перед серверами и направляющего запросы клиентов на все серверы, способные выполнить эти запросы таким образом, чтобы максимально увеличить скорость и загрузку мощностей и не допустить перегрузки одного сервера, что может снизить производительность. Если один из серверов выходит из строя, балансировщик нагрузки перенаправляет трафик на оставшиеся работающие серверы. Для выполнения своих функций балансировщик нагрузки использует различные методы балансировки. [2]

Целью данной работы является классификация методов балансировки высоконагруженных систем.

Для достижения поставленной цели решаются следующие задачи: Рассматриваются известные подходы к балансировке высоконагруженных систем; Описываются методы балансировки высоконагруженных систем, относящиеся к каждому из подходов; Предлагаются и обосновываются критерии оценки качества описанных методов; Сравниваются методы по предложенным критериям оценки; Выделяются методы, показывающие лучшие результаты по одному или нескольким критериям.

В данной работе будут рассмотрены следующие методы балансировки нагрузки: **Source IP-Hash, URL-Hash, Fixed Weighting**.

2 Аналитический раздел

Алгоритмы балансировки нагрузки принято разделять на две категории: **статические** и **динамические**.

В **статических** алгоритмах балансировки нагрузки решения о распределении нагрузки между узлами системы принимаются в начале выполнения, они не зависят от текущего состояния системы. Цель статической балансировки нагрузки – снизить общее время выполнения синхронной программы, минимизируя коммуникационные задержки. [3]

В **динамических** алгоритмах балансировки нагрузки решения о распределении нагрузки принимаются на основе текущего состояния системы, предварительных сведений о системе для этого не требуется. Главное преимущество динамической балансировки нагрузки заключается в том, что отказ одного из узлов не приведёт к остановке системы, а лишь повлияет на ее производительность. [3]

2.1 Source IP-Hash

Source IP-Hash – динамический алгоритм балансировки нагрузки, который объединяет исходный и целевой IP-адреса клиента и сервера для создания уникального хеш-ключа. Ключ используется для привязки клиента к определенному серверу. Поскольку ключ может быть повторно сгенерирован, если сеанс прерван, запрос клиента направляется на тот же сервер, который он использовал ранее. Это полезно, если важно, чтобы клиент подключился к сеансу, который все еще активен после отключения. [1]

2.2 URL-Hash

URL-Hash – динамический алгоритм балансировки нагрузки, который так же, как и **Source IP-Hash** использует хеш-ключ для маршрутизации. Отличие состоит лишь в том, что для генерации хеш-ключа используется URL-адрес интернет-ресурса, к которому происходит обращение. [4]

2.3 Fixed Weighting

Fixed Weighting – динамический алгоритм балансировки нагрузки, при котором администратор назначает вес каждому серверу приложений на основе критериев по их выбору, чтобы продемонстрировать способность серверов приложений обрабатывать трафик. Сервер приложений с наибольшим весом получит весь трафик. Если сервер приложений с наибольшим весом выходит из строя, весь трафик будет направлен на следующий сервер приложений с наибольшим весом. [1]

Список использованных источников

- [1] ИССЛЕДОВАНИЕ И ОБЗОР МЕХАНИЗМОВ БАЛАНСИРОВКИ НАГРУЗКИ НА ОСНОВЕ SDN В 5G/IMT-2020. — Режим доступа: <https://cyberleninka.ru/article/n/issledovanie-i-obzor-mehanizmov-balansirovki-nagruzki-na-osnove-sdn-v-5g-imt-2020>. (Дата обращения: 07.10.2023).
- [2] What is Load Balancing? — Режим доступа: <https://www.nginx.com/resources/glossary/load-balancing>. (Дата обращения: 05.10.2023).
- [3] Load Balancing In Cloud Computing: A Review. — Режим доступа: <https://www.iosrjournals.org/iosr-jce/papers/Vol15-issue2/C01522229.pdf>. (Дата обращения: 07.10.2023).
- [4] World Wide Web Caching: Trends и Techniques. — Режим доступа: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=3104c7c7b7cf44ad4d4e542684de2ec070aa0ba8>. (Дата обращения: 07.10.2023).