ВИДЫ И ЗАДАЧИ МЕТОДОВ И АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ

Д.А. Хлебест, Е.С Омелюсик
Белорусский государственный университетинформатики и радиоэлектроники
г. Минск, Республика Беларусь
Шаталова В.В.–канд.техн.наук, доцент

Развитие и внедрение в повседневную жизнь населения информационных систем обусловило резкое увеличение числа запросов на обработку, увеличение нагрузки на обрабатывающее оборудование (серверы).

Использование высоконагруженных систем, услугами которых пользуется большое количество пользователей, требует применения в качестве аппаратной платформы серверных групп или кластеров. Кластер состоит из нескольких компьютеров, объединенных высокоскоростным соединением. Для пользователей кластер выглядит как один компьютер, а внутри он является разновидностью сети, которая может быть распределенной или локальной.

Важное звено кластера сосредоточено в устройстве или программном обеспечении, распределяющем нагрузку (поток запросов) между компьютерами кластера. Это устройство (программное обеспечение) называется балансировщиком нагрузки. Основной проблемой, связанной с балансировкой, является вопрос о том, как распределять нагрузку наиболее эффективно. Для этого нужно формализировать методику оценки качества работы балансировщика, которая зависит от параметров системы и параметров входного потока запросов.

Поскольку нагрузка на информационные системы будет постоянно расти, задачи балансировки будут приобретать все более важное значение для повышения эффективности информационных систем.

В терминологии компьютерных сетей **балансировка нагрузки** или выравнивание нагрузки (англ. *load balancing*) — метод распределения заданий между несколькими сетевыми устройствами (например, серверами) с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов, горизонтального масштабирования кластера (динамическое добавление/удаление устройств), а также обеспечения отказоустойчивости (резервирования)[1].

В компьютерах **балансировка нагрузки** распределяет нагрузку между несколькими вычислительными ресурсами, такими как компьютеры, компьютерные кластеры, сети, центральные процессоры или диски. Цель балансировки нагрузки — оптимизация использования ресурсов, максимизация пропускной способности, уменьшение времени отклика и предотвращение перегрузки какого-либо одного ресурса. Использование нескольких компонентов балансировки нагрузки вместо одного компонента может повысить надежность и доступность за счет резервирования. Балансировка нагрузки предполагает обычно наличие специального программного обеспечения или аппаратных средств, таких как многоуровневый коммутатор или система доменных имен, как серверный процесс.

Балансировка нагрузки отличается от физического соединения тем, что балансировка нагрузки делит трафик между сетевыми интерфейсами на сетевой сокет (модель OSI уровень 4) основе, в то время как соединение канала предполагает разделение трафика между физическими интерфейсами на более низком уровне, либо в пакет (модель OSI уровень 3) или по каналу связи (модель OSI уровень 2).

Примеры устройств, к которым применима балансировка:

- серверные кластеры;
- прокси-серверы;
- межсетевые экраны;
- коммутаторы;
- серверы инспектирования содержимого;
- серверы *DNS*;
- сетевые адаптеры.

Балансировка нагрузки может быть использована для расширения возможностей фермы серверов, состоящей более чем из одного сервера. Она также может позволить продолжать работу даже в условиях, когда несколько исполнительных устройств (серверов) вышли из строя. Благодаря этому растёт отказоустойчивость, и появляется возможность динамически регулировать используемые вычислительные ресурсы за счёт добавления/удаления исполнительных устройств в кластере.

Существует множество **методов и алгоритмов**, которые можно использовать для интеллектуальной **балансировки нагрузки** запросов клиентов на пулы серверов [2]. Выбранный метод будет зависеть от типа обслуживаемой услуги или приложения, а также от состояния сети и серверов на момент запроса. Методы, описанные ниже, будут использоваться в комбинации, чтобы определить лучший сервер для обслуживания новых запросов. Текущий уровень запросов к балансировщикам нагрузки часто определяет, какой метод используется. Когда нагрузка мала, тогда будет достаточно одного из простых методов балансировки нагрузки. В периоды высокой нагрузки используются более сложные методы для обеспечения равномерного распределения запросов [3].

Основной задачей балансировщика является определение того, какой из серверов из бэкенд-пула может наиболее эффективно обработать входящий пакет данных. Для этого у балансировщика (Load Balancer, LB) имеется несколько алгоритмов, наличие и возможность применения которых зависит от типа LB и его настроек [4, 5].

Группы балансировки нагрузки используют алгоритмы для принятия решений о распределении нагрузки. Решение определяет, какому удаленному серверу переслать новое соединение.

Группы балансировки нагрузки поддерживают взвешенные и невзвешенные алгоритмы.

Взвешенный алгоритм использует вес (или предпочтение), чтобы помочь определить, какой сервер получает следующий запрос. Сервер с большим весом получает больше трафика, чем сервер с меньшим весом. Процент трафика на каждый сервер приблизительно равен его весу, деленному на совокупный вес всех серверов в группе.

Невзвешенный алгоритм предполагает, что емкость всех серверов в группе эквивалентна. Хотя невзвешенные алгоритмы обычно быстрее, чем взвешенные алгоритмы, некоторые невзвешенные алгоритмы, такие как алгоритм хеширования, могут отправлять больше трафика на некоторые серверы. Если в группе есть серверы с разными мощностями, обработка не может оптимизировать мощности всех серверов [6].

Выбирая конкретный алгоритм, нужно исходить, во-первых, из специфики конкретного проекта, а во-вторых – из целей, которые мы планируем достичь.

В числе целей, для достижения которых используется балансировка, нужно выделить следующие:

– справедливость: нужно гарантировать, чтобы на обработку каждого запроса выделялись системные ресурсы и не допустить возникновения ситуаций, когда один запрос обрабатывается, а все остальные ждут своей очереди;

– эффективность: все серверы, которые обрабатывают запросы, должны быть заняты на 100%; желательно не допускать ситуации, когда один из серверов простаивает в ожидании запросов на обработку (в реальной практике эта цель достигается далеко не всегда);

– сокращение времени выполнения запроса: нужно обеспечить минимальное время между началом обработки запроса (или его постановкой в очередь на обработку) и его завершения;

– сокращение времени отклика: нужно минимизировать время ответа на запрос пользователя.

Очень желательно также, чтобы алгоритм балансировки обладал следующими свойствами:

– предсказуемость: нужно чётко понимать, в каких ситуациях и при каких нагрузках алгоритм будет эффективным для решения поставленных задач;

– равномерная загрузка ресурсов системы;

– масштабируемость: алгоритм должен сохранять работоспособность при увеличении нагрузки.

Список используемых источников:

1. Балансировка нагрузки [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Балансировка_нагрузки. – Дата доступа: 13.01.2021.

2. Weighted Least-Connection Scheduling [Электронный ресурс]. – Режим доступа: http://kb.linuxvirtualserver.org/wiki/Weighted_Least-Connection_Scheduling. – Дата доступа: 13.01.2021.

3. Algorithms for making load-balancing decisions [Электронный ресурс]. – Режим доступа: https://www.ibm.com/support/knowledgecenter/SS9H2Y_7.6.0/com.ibm.dp.doc/lbg_algorithms.html. – Дата доступа: 13.01.2021.

4. Comparing Load Balancing Algorithms [Электронный ресурс]. – Режим доступа: https://www.jscape.com/blog/load-balancing-algorithms. – Дата доступа: 13.01.2021.

5. Алгоритмы и методы балансировки нагрузки [Электронный ресурс]. – Режим доступа: https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques. – Дата доступа: 13.01.2021.

6. Задача балансировки нагрузки на серверы [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/zadacha-balansirovki-nagruzki-na-servery/viewer. – Дата доступа: 13.01.2021.