

1 Анализ предметной области

1.1 Динамическая балансировка

Динамические алгоритмы осуществляют мониторинг состояния каждого из узлов и выбирают «наилучший», в рассматриваемый момент времени, из них [1]. К динамическим алгоритмам относятся:

- Least Connections
- Weighted Least Connections
- Least Response Time

1.1.1 Least Connections

Алгоритм Least Connections распределяет нагрузку между узлами, в зависимости от количества активных соединений, обслуживаемых каждым узлом. Узел с наименьшим числом соединений будет обрабатывать следующий запрос, а узлы с большим числом соединений будут перераспределять свою нагрузку на узлы с меньшей загрузкой [2].

Пусть имеется N запросов и M узлов. Пусть для каждого узла есть количество активных соединений *conns*. Алгоритм состоит из следующих шагов:

1. сформировать массив, содержащий узлы;
2. установить указатель *target* на первый узел;
3. пройти циклом по всем узлам массива, кроме первого:
 - если *conns* текущего узла меньше *conns* узла *target*, то установить *target* на текущий узел;
4. отправить запрос на узел *target*.

Особенности алгоритма Least Connections:

- низкая степень прогнозирования;
- низкая стабильность;

- высокая отказоустойчивость, поскольку постоянно собирается информация об узлах, и, в случае отказа, система перераспределит ресурсы;
- высокая потребность в ресурсах, поскольку необходимо постоянно собирать информацию о узлах в реальном времени;
- высокое время обработки нового запроса, поскольку балансировщику нагрузки необходимо время, чтобы правильно перенаправить задачу;
- если *conns* у каждого узла одинаково, то алгоритм следует выбору по правилам алгоритма Round Robin.

1.1.2 Weighted Least Connections

Данный алгоритм комбинирует принципы алгоритмов Least Connections и Weighted Round Robin [3]. Он учитывает как веса узлов, так и количество активных соединений. Новое сетевое подключение предоставляется узлу, который имеет минимальное отношение количества текущих активных подключений к его весу [4].

Пусть имеется N запросов и M узлов. Пусть для каждого узла есть количество соединений *conns* и вес *weight*. Алгоритм состоит из следующих шагов:

1. сформировать массив, содержащий узлы;
2. установить указатель *target* на первый узел;
3. пройтись циклом по всем узлам массива, кроме первого:
 - если отношение *conns* и *weight* текущего узла меньше отношения *conns* и *weight* узла *target*, то установить *target* на текущий узел;
4. отправить запрос на узел *target*.

Особенности алгоритма Weighted Least Connections:

- динамический тип;
- низкая степень прогнозирования;
- низкая стабильность;

- высокая отказоустойчивость;
- высокая потребность в ресурсах;
- высокое время обработки нового запроса;
- благодаря весам, возможно осуществить настройку алгоритма таким образом, чтобы он учитывал различия в технических характеристиках узлов.
- при учитывании сразу двух критериев (количества подключений и веса) выбор нового узла происходит более сбалансировано.

1.1.3 Least Response Time

Данный алгоритм имеет схожесть с алгоритмом Least Connections, только при распределении нагрузки он руководствуется наименьшим временем ответа узла. При выборе учитывается производительность узлов и балансировщик стремится направить запрос к наиболее подходящему узлу [5].

Пусть имеется N запросов и M узлов. Пусть для каждого узла есть время ответа на предыдущий запрос $time$. Алгоритм состоит из следующих шагов:

1. сформировать массив, содержащий узлы;
2. установить указатель $target$ на первый узел;
3. пройти цикл по всем узлам массива, кроме первого:
 - если $time$ текущего узла меньше $time$ узла $target$, то установить $target$ на текущий узел;
4. отправить запрос на узел $target$.

Особенности алгоритма Least Response Time:

- динамический тип;
- низкая степень прогнозирования;
- низкая стабильность;

- высокая отказоустойчивость;
- высокая потребность в ресурсах;
- высокое время обработки нового запроса;
- если время ответа каждого узла одинаково, то алгоритм следует выбору по правилам алгоритма Round Robin.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Управление трафиком в сети с высокой динамикой метрик сетевых маршрутов. — Режим доступа: <https://cyberleninka.ru/article/n/upravlenie-trafikom-v-seti-s-vysokoy-dinamikoy-metrik-setevyh-marshrutov?ysclid=lqwlxbfbif183638987>.
2. Husain N., Timotius W. Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler // Informatika: Jurnal Teknologi Komputer dan Informatika. — 2016. — Т. 12, № 1.
3. Павликов М. К. Алгоритм распределения нагрузки в программной системе, построенной на основе протокола HDP // Вестн. Том. гос. ун-та. Управление, вычислительная техника и информатика. — 2017. — № 40.
4. An Improved Weighted Least Connection Scheduling Algorithm for Load Balancing in Web Cluster Systems [Электронный ресурс]. — Режим доступа: <https://www.irjet.net/archives/V5/i3/IRJET-V5I3455.pdf/> (дата обращения: 06.10.2023).
5. System design: Load Balancing. — <https://pipscrip.hashnode.dev/system-design-load-balancing> (дата обращения: 20.12.2023).