

ИУ7-54Б, 16\_KOZ, Турчанский

## ОПРЕДЕЛЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие термины с соответствующими определениями.

Балансировщик нагрузки — это устройство, которое находится между пользователем и группой серверов и действует как невидимый посредник, обеспечивая одинаковое использование всех серверов ресурсов.[1]

Время ответа — это общее время, затрачиваемое сервером на обработку входящих запросов и отправку ответа.[1]

Вес — вероятность, с которой балансировщик нагрузки в следующий раз выберет этот узел [2].

Вычислительный узел (узел) — устройство, выполняющее основную логику обработки запроса [3] Распределенная система - это вычислительная среда, в которой различные компоненты распределены между несколькими компьютерами (или другими вычислительными устройствами)

# ВВЕДЕНИЕ

У Мишани введение норм и так сойдет

Цель работы — исследование методов балансировки высоконагруженных систем.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- провести анализ предметной области балансировки нагрузки высоконагруженных систем;
- провести обзор существующих методов балансировки нагрузки высоконагруженных систем;
- сформулировать критерии сравнения методов балансировки нагрузки высоконагруженных систем;
- провести сравнение методов балансировки нагрузки высоконагруженных систем;
- сформулировать выводы.

## 1 Анализ предметной области

Балансировка нагрузки — это механизм приблизительного выравнивания рабочей нагрузки между всеми узлами распределенной системы [4]. Это позволяет ускорить обслуживание заданий, например, свести к минимуму время отклика и повысить эффективность использования ресурсов.

### 1.1 Алгоритмы решения задачи балансировки

Балансировщик нагрузки работает по одному из алгоритмов, решающих задачу балансировки. На вход этому алгоритму подается некоторое число запросов, приходящих в систему и набор вычислительных узлов, которыми располагает система. Задача алгоритма сводится к минимизации времени обработки запросов, за счет распределения запросов по вычислительным узлам.

Для анализа алгоритмов балансировки могут быть выделены следующие параметры [3]:

- точность прогнозирования — степень соответствия расчетных результатов работы алгоритма их фактическому значению;
- стабильность — уровень задержек в передаче информации между всеми участниками балансировки;
- отказоустойчивость — показывает устойчивость алгоритма к возникновению разнообразных ошибок;
- потребность в ресурсах — количество ресурсов, затрачиваемых на процесс балансировки нагрузки;
- время обработки нового запроса — время от поступления нового запроса до его перенаправления к цели.

Алгоритмы балансировки разделяют на статические и динамические [3].

### 1.2 Статическая балансировка

Статическая балансировка — это метод распределения нагрузки на узлы, основанный на заранее определенных параметрах.

При использовании статической балансировки узлы настраиваются по разному [5]:

- По равномерному распределению нагрузки — каждый узел получает аналогичное количество запросов.
- По приоритету — выделенные узлы имеют больший приоритет и получают больше запросов.
- По типу запроса — различные типы запросов могут быть отправлены на разные узлы для оптимизации обработки.

### 1.3 Динамическая балансировка

Динамические алгоритмы осуществляют мониторинг состояния каждого из узлов и выбирают «наилучший», в рассматриваемый момент времени, из них [6]. К динамическим алгоритмам относятся:

- Least Connections
- Weighted Least Connections
- Least Time

#### 1.3.1 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 [7].

Пусть имеется  $N$  запросов и  $M$  узлов. Пусть для каждого узла есть количество активных соединений *conn<sub>s</sub>*. 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:

- динамический тип;
- низкая степень прогнозирования;
- низкая стабильность;
- высокая отказоустойчивость;
- высокая потребность в ресурсах;
- высокое время обработки нового запроса.

### 1.3.2 Weighted Least Connections

<https://oaji.net/articles/2019/2698-1569231711.pdf>

Данный алгоритм комбинирует принципы алгоритмов Least Connections и Weighted Round Robin [2]. 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 [5].

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. пройти циклом по всем узлам массива, кроме первого:
  - если соотношение *conns* и *weight* текущего узла выгоднее соотношения *conns* и *weight* узла *target*, то установить *target* на текущий узел;

4. отправить запрос на узел *target*.

<https://cyberleninka.ru/article/n/sravnitelnyy-analiz-algoritmov-balansirovki-nagruzki-v-srede-oblachnyh-vychisleniy/viewer>

Особенности алгоритма Weighted Least Connections:

- динамический тип;
  - низкая степень прогнозирования;
  - низкая стабильность;
  - высокая отказоустойчивость;
  - высокая потребность в ресурсах;
  - высокое время обработки нового запроса.
1. сформировать массив, содержащий узлы;
  2. установить указатель *target* на первый узел;
  3. пройти цикл 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*.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое балансировка нагрузки? [Электронный ресурс]. — Режим доступа: <https://aws.amazon.com/ru/what-is/load-balancing/> (дата обращения: 05.10.2023).
2. Алгоритм распределения нагрузки в программной системе, построенной на основе протокола HDP [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/algoritm-raspredeleniya-nagruzki-v-programmnoy-sisteme-postroennoy-na-osnove-protokola-hdp> (дата обращения: 06.10.2023).
3. СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-algoritmov-balansirovki-nagruzki-v-srede-oblachnyh-vychisleniy> (дата обращения: 07.10.2023).
4. РУКОВОДСТВО ПО ДИНАМИЧЕСКОЙ БАЛАНСИРОВКЕ НАГРУЗКИ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/rukovodstvo-po-dinamicheskoy-balansirovke-nagruzki-v-raspredelennyh-kompyuternyh-sistemah?ysclid=lqwjcuy2ch543400454> (дата обращения: 06.10.2023).
5. 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).
6. 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>.
7. Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler [Электронный ресурс]. — Режим доступа: <https://www.neliti.com/publications/67705/analisis-algoritma-round-robin-least-connection-dan-ratio-pada-load-balancing-me> (дата обращения: 06.10.2023).