

УДК 004.021:519.687.1

## РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ МЕТОДА «ВЗВЕШЕННЫХ НАИМЕНЬШИХ СОЕДИНЕНИЙ» БАЛАНСИРОВКИ НАГРУЗКИ НА СЕРВЕРЫ

*Хлебест Д.А., Омелюсик Е.С.*

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

*Научный руководитель: Шаталова В.В. – канд.техн.наук, доцент*

**Аннотация.** Разработана математическая модель модели метода балансировки нагрузки на серверы, позволяющая распределять нагрузку наиболее эффективно, повышая качество балансировки путём выявления и устранения недостатков присущих известным методам и алгоритмам.

**Ключевые слова:** балансировка нагрузки, распределение трафика, программное управление

**Введение.** 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 [1, 2, 3].

Важное звено кластера сосредоточено в устройстве или программном обеспечении (ПО), распределяющем нагрузку (поток запросов) между серверами. Это устройство или ПО называется балансировщиком нагрузки. Основной проблемой является наиболее эффективное распределение нагрузки. Для этого нужно формализовать методику оценки качества работы балансировщика, зависящей 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 [4].

**Основная часть.** Метод «Взвешенных наименьших соединений» поддерживает взвешенный список серверов приложений с их числом активных соединений. Служба переадресует новое соединение с сервером на основе следующей комбинации: его пропорция к весу или предпочтению, количество активных подключений. Это приводит к более эффективному распределению трафика на сервер, который наиболее способен обработать запрос [5].

Метод имеет различные реализации в зависимости от требований и дополнительных параметров, влияющих на расчёт приоритета. Каждый разработчик/команда/компания разрабатывает алгоритм с нуля и только для себя. В открытом доступе реализаций алгоритма WLC нет, пусть он гипотетически имеет наибольшую эффективность в большинстве случаев.

Этот алгоритм использует больше времени вычислений, чем алгоритм наименьшего соединения, однако дополнительные вычисления приводят к более эффективному распределению трафика на сервер [6].

В алгоритме взвешенного наименьшего планирования соединений (*Weighted Least Connections, WLC*) каждому серверу может быть присвоен различный вес производительности. Алгоритм планирования взвешенного наименьшего количества соединений делает алгоритм наименьшего количества соединений тем же, что алгоритм взвешенного циклического

перебора для алгоритма округлого, то есть он вводит «вес», основанный на спецификациях каждого сервера [7].

Балансировщик нагрузки, который реализует алгоритм *WLC*, теперь учитывает две вещи: вес (ёмкость каждого сервера) и текущее количество клиентов, в настоящее время подключенных к каждому серверу.

Алгоритм может быть применён в «диспетчере» – распределителе нагрузки на серверы и единой точке входа в распределённую систему, для последующего распределения клиентов на различные *API*-шлюзы (*Application Programming Interface*, программный интерфейс приложения). Схема работы диспетчера распределителя нагрузки представлена на рисунке 1.

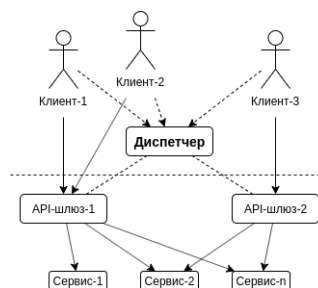


Рисунок 1 – Схема работы диспетчера распределителя нагрузки

Формула расчёта веса для «*API*-шлюза»:

$$W = \sqrt{(\rho_{mem} \cdot Q_{mem} \cdot R_{mem})^2 \cdot (\rho_{cpu} \cdot Q_{cpu} \cdot R_{cpu})^2}, \quad (1)$$

где  $R_{mem}$  – простой памяти;  
 $R_{cpu}$  – простой процессора;  
 $Q_{mem}$  – объём памяти (Кб);  
 $V_{cpu}$  – скорость процессора (МГц);  
 $\rho_{mem}$  и  $\rho_{cpu}$  – коэффициенты пропорциональности.  
 Формула расчёта приоритета для «*API*-шлюза»:

$$P_i = \frac{C_i}{W_i}, \quad (2)$$

где  $n$  – количество узлов;  
 $W_i$  – веса узлов ( $i = 1..n$ );  
 $C_i$  – количество текущих подключений ( $i = 1..n$ ).  
 Схематичная реализация алгоритма. Сокращённые обозначения объектов и параметров:  
 –  $[O]$  – оптимальный (следующий) «*API*-шлюз»;  
 –  $[C]$  – количество текущих подключений «*API*-шлюза»;  
 –  $[W]$  – вес «*API*-шлюза»;  
 –  $[P]$  – приоритет «*API*-шлюза»;  
 –  $[L]$  – любая дополнительная текстовая информация о «*API*-шлюзе».

Взаимодействие с сервисом «Диспетчера» сводится к 4-м операциям CRUD (создание, чтение, обновление, удаление):

1. Запрос на создание инициирует новый «*API*-шлюз», готовый принимать на себя последующие подключения клиентов. В запросе он должен передать начальное количество подключений и параметры для расчёта веса. В ответ он получит личный уникальный идентификатор во всей системе.

- Формат запроса:  $\{ P: \{ C, W: \{ R_{mem}, R_{cpu}, Q_{mem}, V_{cpu} \} \}, I: \{ \dots \} \}$
- Формат ответа  $\rightarrow \{ id \}$

2. Запрос на обновление инициирует уже работающий «API-шлюз» по событию изменения числа обрабатывающих подключений (подключение или отключение клиентов). В запросе он должен передать новое число открытых соединений. В ответ он получит статус выполнения запроса.

- a. Формат запроса:  $\{ C \}$
- b. Формат ответа  $\rightarrow \{ \dots \}$

3. Запрос на удаление инициирует «API-шлюз», завершающий работу по каким-либо причинам. В запросе и ответе параметры не требуются.

- a. Формат запроса:  $\{ \dots \}$
- b. Формат ответа  $\rightarrow \{ \dots \}$

4. Запрос на чтение инициирует новый клиент, желающий получить начать работу с распределённой системой. Для выполнения запроса параметры не требуются. В ответ он получит оптимальный «API-шлюз» с минимальной нагрузкой.

- a. Формат запроса:  $\{ \dots \}$
- b. Формат ответа  $\rightarrow \{ O \}$

Для оптимального выполнения вышеописанных запросов, сервис «Диспетчера» в памяти должны кэшироваться «активный пул» соединений с форматом: **activePool**:  $\{ \text{"id"}: \{ P_1: \{ C_1, W_1 \}, I_1: \{ \dots \} \}, \dots \}$ , а также переменная-объект, содержащая оптимальный на текущий момент «API-шлюз»: **optimalChoice**:  $\{ id, P, I \}$ .

**Заключение.** Разработан новый контентно-независимый алгоритм распределения нагрузки для распределённых систем, учитывающий как количество текущих соединений, так и мощность серверов, по методу «Взвешенных наименьших соединений». Алгоритм взвешенных наименьших соединений один из самых распространённых выборов при разработке больших проектов с распределёнными системами, реализация которого варьируется от особенностей системы, а разработка требует наиболее оптимального решения. Полученные результаты могут быть полезны разработчикам и администраторам вычислительных систем и сетей при управлении нагрузкой систем клиент-сервер.

### Список литературы

1. Либман Л. Философия распределения нагрузки // Журнал сетевых решений LAN, 2001, с. 350.
2. Герасимов А.И. Оптимизация и балансировка систем и сетей с учетом поступающей информации. Труды института конструкторского-технологической информатики РАН. М., 2005.
3. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб. Литер, 2003. - 877с.
4. Богуславский Л.Б., Ляхов А.И. Оценка производительности распределенных информационно-вычислительных систем архитектуры "клиент-сервер". Автоматика и телемеханика, №9, 1995.
5. Бакусов Л.М. и др. Математические модели информационных процессов и управления в АСУ: Учебн. пособие. Уфа, 1991.
6. Алгоритмы принятия решения по балансировке нагрузки [Электронный ресурс]. – Режим доступа: [https://www.ibm.com/support/knowledgecenter/SS9H2Y\\_7.6.0/com.ibm.dp.doc/lbg\\_algorithms.html](https://www.ibm.com/support/knowledgecenter/SS9H2Y_7.6.0/com.ibm.dp.doc/lbg_algorithms.html). – Дата доступа: 25.11.2019.
7. S.Sharma, S.Singh, M.Sharma. Анализ производительности алгоритмов балансировки нагрузки: всемирная академия наук, инженерии и технологии, 2008.

UDC 004.021:519.687.1

## DEVELOPMENT OF A MATHEMATICAL MODEL OF THE "WEIGHTED-LOWEST CONNECTIONS" METHOD FOR BALANCING LOADING ON SERVERS

Khlebest D.A., Omelyusik E.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Shatalova V.V. – PhD, associate professor

**Annotation.** A mathematical model of the method of balancing the load on servers has been developed, which allows distributing the load most efficiently, increasing the quality of balancing by identifying and eliminating the shortcomings inherent in known methods and algorithms.

**Keywords.** load balancing, traffic distribution, software control