На правах рукописи

Викулов Егор Олегович

МЕТОДЫ И АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ НАГРУЗКИ МЕЖДУ ВЫЧИСЛИТЕЛЬНЫМИ РЕСУРСАМИ ИНФОРМАЦИОННЫХ СИСТЕМ

Специальность 2.3.1. Системный анализ,
управление и обработка информации, статистика

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Омск – 2024

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Омский государственный технический университет» (ОмГТУ).

Научный руководитель:	**Денисова Людмила Альбертовна,** доктор технических наук, доцент.
Официальные оппоненты:	**Авдеенко Татьяна Владимировна,** доктор технических наук, профессор, профессор кафедры теоретической и прикладной информатики федерального государственного бюджетного образовательного учреждения высшего образования «Новосибирский государственный технический университет», г. Новосибирск; **Сервах Владимир Вицентьевич,** доктор физико-математических наук, старший научный сотрудник, старший научный сотрудник Омского филиала федерального государственного бюджетного учреждения науки «Институт математики им. С.Л. Соболева Сибирского отделения Российской академии наук», г. Омск.
Ведущая организация:	Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова», г. Барнаул.

Защита состоится «26» июня 2024 г. в 15-00 на заседании диссертационного совета 24.2.350.07, созданного на базе ФГАОУ ВО «Омский государственный технический университет», по адресу: 644050, Омск, просп. Мира, д. 11, Главный корпус, ауд. П-202.

С диссертацией можно ознакомиться в библиотеке ФГАОУ ВО «Омский государственный технический университет» и на сайте www.omgtu.ru.

Отзыв на автореферат в двух экземплярах, заверенный печатью учреждения, просьба направлять по адресу: 644050, г. Омск, пр. Мира, д. 11, ученому секретарю диссертационного совета 24.2.350.07. Тел.: (3812) 65-24-79, e-mail: dissov_omgtu@omgtu.ru.

Автореферат разослан «____» _____ 2024 года.

Ученый секретарь диссертационного совета
24.2.350.07, канд. техн. наук, доцент  Грицай А.С.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. На сегодняшний день наблюдается стремительный рост числа пользователей сети интернет, что приводит к возрастанию нагрузки на клиент-серверные информационные системы. Пользователи ожидают, что информационные системы будут работать быстро и без сбоев. Всё больше клиентов используют мобильные устройства для доступа в интернет, что также приводит к увеличению нагрузки. В свою очередь, клиент-серверные информационные системы предоставляют все больше информации, более качественный и высокоскоростной сервис, чтобы привлечь и удержать клиентов. Повышение требований пользователей к качеству обслуживания приводит к необходимости привлечения дополнительных ресурсов для обработки и хранения данных. Поэтому важным является учет состояния серверов в вычислительном комплексе. Для распределения нагрузки между узлами (облачными ресурсами или физическими серверными станциями) часто используется механизм балансировки. Термин «балансировка нагрузки» (англ. *load balancing*) используется для обозначения как процесса перенаправления данных, получаемых от клиентов, на определённый узел системы, так и распределения вычислительной нагрузки приложений и информационных систем. Следует отметить, что в настоящее время в России действуют законодательные требования к обеспечению доступности государственных услуг для пользователей в электронном виде (в частности требования Федерального закона №152 «О персональных данных»). Выполнение этих требований позволяет обеспечить применение балансировки нагрузки между ресурсами. Таким образом, задача распределения нагрузки в клиент-серверных информационных системах в настоящее время является актуальной.

Состояние вопроса. Разработке методов и алгоритмов распределения нагрузки клиент-серверных информационных систем посвящены труды следующих отечественных и зарубежных ученых: Лохвицкого В.А., Гончаренко В.А., Никишина К.И., Sahoo J., Salahuddin M. A., Glitho R., Elbiaze H. Интеллектуальным технологиям обработки данных (на основе кластерного анализа и аппарата нечеткой логики) посвящены труды таких ученых как Каллан Р., Kohonen T., Lakhmi C. Jain, Zadeh L. A., Штовба С.Д., Хайкин С. Исследования в

области анализа паттернов данных представлены в работах Алескерова Ф.Т., Андрейчикова А.В., Few S., и других авторов в России и за рубежом.

Анализ существующих методов и алгоритмов балансировки нагрузки показал, что они не учитывают или учитывают не в полной мере состояние вычислительных ресурсов. Поэтому перспективным представляется развитие методов и алгоритмов балансировки, в том числе на основе интеллектуальных технологий обработки данных о состоянии ресурсов, что позволило бы значительно повысить качество информационного обслуживания пользователей.

Целью диссертационной работы является повышение быстродействия и производительности высоконагруженных клиент-серверных информационных систем путем оптимизации распределения вычислительной нагрузки и статических данных между серверами. Для достижения указанной цели в работе поставлены и решены следующие **задачи**:

1. Анализ проблемы повышения производительности работы клиент-серверных информационных систем.

2. Обоснование показателей состояния вычислительных ресурсов на основе паттерн-анализа функционирования серверов для выбора сервера при распределении нагрузки.

3. Разработка методов и алгоритмов на основе кластерного анализа и нечеткого логического вывода, обеспечивающих повышение быстродействия и производительности систем распределения нагрузки между серверами в сравнении с известными методами в условиях неполноты данных о состоянии вычислительных ресурсов.

4. Разработка структуры программного комплекса и алгоритма параллельных вычислений для балансировки нагрузки, а также проведение экспериментальных исследований для оценки эффективности распределения данных между ресурсами облачного серверного кластера.

Научная новизна. В процессе исследований получены следующие новые научные результаты.

1. Разработан комбинированный метод формирования показателей и правил выбора сервера при балансировке нагрузки на основе данных о состоянии серверного комплекса. Отличительными особенностями метода является

выделение паттернов показателей функционирования серверов, позволяющее определить значимые критерии выбора, влияющие на скорость обработки запросов пользователей, а также последующая кластеризация запросов для формирования правил выбора сервера.

2. Предложен аналитико-имитационный метод распределения вычислительной нагрузки с помощью нечеткого логического вывода, положенного в основу работы сервера-балансира. В отличие от существующих метод, включающий аналитическую обработку экспериментальных данных о состоянии вычислительных ресурсов в совокупности с модельными исследованиями, позволяет выбрать и обосновать параметры алгоритма балансировки, обеспечивая повышение быстродействия высоконагруженной информационной системы.

3. Разработан алгоритм параллельных вычислений для распределения данных между ресурсами облачного серверного кластера и структура программного комплекса для проведения экспериментальных исследований. Основным преимуществом данного алгоритма в отличие от существующих является возможность выполнения больших объемов вычислений в параллельном потоке, что повышает скорость доставки данных. Отличительной особенностью структуры программного комплекса является декомпозиция системы балансировки на параллельно работающие программы (с выделением модуля принятия решений о выборе сервера), что существенно увеличивает быстродействие системы и позволяет проводить эксперименты с облачными ресурсами.

Практическая значимость работы заключается в разработке:

– программного комплекса, метода и алгоритма сбора и обработки параметров состояния серверов, которые позволяют выявить закономерности в данных и обосновать правила выбора сервера для распределения нагрузки.

– аналитико-имитационного метода, позволяющий проводить тестирование и имитационное моделирование балансировки вычислительной нагрузки по серверам.

– программного комплекса и алгоритма параллельных вычислений распределения нагрузки по серверам.

Внедрение результатов исследований. Созданные программные комплексы сбора данных и распределения вычислительной нагрузки использованы в

разработке программных продуктов ООО «РОНАС ИТ», что позволило повысить скорость работы, стабильность и отказоустойчивость разрабатываемых высоконагруженных информационных систем. Результаты исследований внедрены в учебный процесс ОмГТУ и используются в учебных дисциплинах кафедры «Автоматизированные системы управления и обработки информации» (АСОИУ).

Объектом исследования являются способы распределения данных и вычислительной нагрузки в высоконагруженных клиент-серверных информационных системах.

Предметом исследования являются математические модели, методы и алгоритмы, предназначенные для анализа и оптимизации распределения нагрузки между вычислительными ресурсами клиент-серверных приложений.

Методология исследования базируется на основах системного анализа, методах анализа больших объемов данных, кластерного анализа, методах нечеткой логики и теории массового обслуживания.

Основные положения, выносимые на защиту.

1. Комбинированный метод формирования параметров и правил выбора сервера при балансировке нагрузки на основе данных о состоянии серверного комплекса. Отличительными особенностями метода является выделение паттернов показателей состояния серверов, позволяющее выявить значимые критерии выбора, влияющие на скорость обработки данных, а также последующая кластеризация запросов пользователей для формирования правил выбора сервера.

2. Аналитико-имитационный метод распределения вычислительной нагрузки с помощью нечеткого логического вывода, положенного в основу работы сервера-балансира. В отличие от существующих метод, включающий аналитическую обработку экспериментальных данных о состоянии вычислительных ресурсов в совокупности с модельными исследованиями, позволяет выбрать и обосновать параметры алгоритма балансировки, обеспечивая повышение быстродействия и отказоустойчивости высоконагруженной информационной системы.

3. Алгоритм параллельных вычислений для распределения данных между ресурсами облачного кластера серверов и структура программного комплекса для проведения экспериментальных исследований. Основным преимуществом данного

алгоритма в отличие от существующих является возможность выполнения больших объемов вычислений в параллельном потоке, что повышает скорость доставки данных. Отличительной особенностью структуры программного комплекса является декомпозиция системы балансировки на параллельно работающие подсистемы (с выделением модуля принятия решений о выборе сервера), что существенно увеличивает быстродействие системы и позволяет проводить эксперименты с облачными ресурсами.

Соответствие паспорту специальности. Диссертация соответствует областям исследований: п. 4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта»; п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта»; п. 13 «Методы получения, анализа и обработки экспертной информации, в том числе на основе статистических показателей».

Достоверность полученных результатов. Обоснованность и достоверность теоретических результатов, положений и выводов, полученных в диссертационной работе, базируются на использовании апробированных научных положений и методов исследования, корректном применении математического аппарата, согласованности новых результатов с известными теоретическими положениями. Обоснованность и достоверность прикладных результатов диссертации подтверждается результатами апробации и внедрения предложенных методов и алгоритмов при проектировании программного комплекса распределения вычислительной нагрузки по серверам.

Апробация результатов исследования. Результаты работы отражались в научных докладах, которые представлялись на: V, VII, VIII, X Всероссийских научно-практических конференциях «Информационные технологии и автоматизация управления» (г. Омск, 2013, 2016, 2017, 2019); на IX, XIII Международных IEEE конференциях «Динамика систем, механизмов и машин», AMSD (г. Омск, 2014, 2019); на II, IV Международных научно-технических конференциях «Проблемы машиноведения», MSTU (г. Омск, 2018, 2020); на

Международной научно-технической конференции «Пром-Инжиниринг» (г. Москва, 2018); на II Международной научно-технической конференции «Научный потенциал молодежи и технический прогресс» (г. Санкт-Петербург, 2019); на Международном семинаре «Передовые технологии в материаловедении, машиностроении и автоматизации», МІР-2019 (г. Красноярск, 2019).

Публикации по теме исследования. По теме диссертации опубликовано 19 научных работ, в том числе 4 научные статьи в рецензируемых научных изданиях, рекомендованных ВАК при Минобрнауки России, 5 научных статей в изданиях, индексируемых в международной реферативной базе данных Scopus, 2 свидетельства о государственной регистрации программ для ЭВМ.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка использованных источников (119 наименований) и трех приложений. Общий объем работы 135 страниц, в том числе 122 страницы основного текста, включая 50 рисунков и 21 таблицу.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, проведен анализ степени разработанности исследуемой научной проблемы и обоснованы подходы к ее решению, поставлена цель работы, сформулированы задачи исследования и основные результаты, выносимые на защиту, указаны научная новизна и практическая значимость работы, а также степень достоверности результатов.

Первая глава посвящена анализу состояния проблемы распределения нагрузки между серверами вычислительного комплекса. Исследованы существующие методы и алгоритмы решения задачи балансировки нагрузки.

Установлено, что актуальным является определение параметров состояния серверов, влияющих на скорость доставки данных конечному пользователю, а также анализ полученных параметров состояния для выполнения дальнейшего распределения запросов пользователей.

Определена необходимость разработки программного обеспечения, выполняющего интеллектуальный анализ больших объемов данных параметров состояния серверов и данных о пользователях, а также программного обеспечения, выполняющего оптимизацию распределения данных между серверами.

Во второй главе представлены результаты сбора и обработки данных о состоянии серверного комплекса. Рассмотрен алгоритм получения данных о состоянии узлов серверного комплекса и процедура формирования правил выбора

сервера. Выявлены и обоснованы параметры состояния серверов пригодные для распределения запросов пользователей. Предложен метод распределения нагрузки между вычислительными ресурсами на основе кластерного анализа параметров состояния серверов.

В работе рассмотрены вопросы балансировки нагрузки в клиент-серверных системах для двух задач: распределения статических данных (РСД) и распределение вычислительной нагрузки (РВН). На рисунке 1 приведена схема клиент-серверного приложения с балансиром нагрузки, реализующим подсистемы распределения статических данных и вычислительной нагрузки.

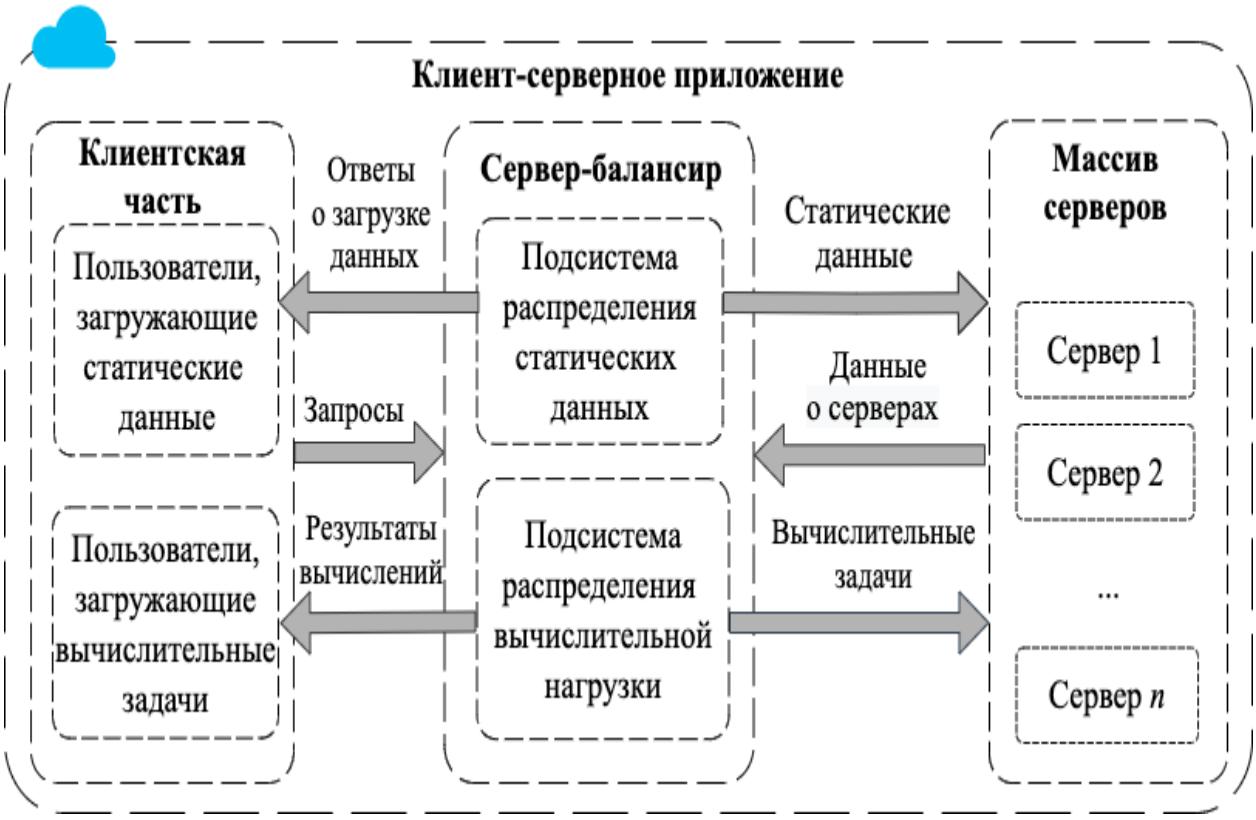

Рисунок 1 – Схема клиент-серверного приложения с балансиром нагрузки.

Для решения первой задачи необходимо распределить по вычислительным ресурсам такие статические данные как текстовые документы, графические файлы, цифровой видеоряд и т.д. для хранения и предоставления по запросу пользователей. Для того, чтобы решить вторую задачу требуется распределить вычислительные запросы по узлам серверного комплекса, а также осуществить поиск меняющейся во времени информации, чтение, добавление или удаление данных и т.д. На рисунке 2 представлен алгоритм сбора и обработки параметров состояния вычислительных ресурсов облачного серверного комплекса. После сбора данных (основной цикл алгоритма) выполняется их обработка,

заключающаяся в анализе паттернов данных о состоянии серверов, кластеризации и построении функций принадлежности для выбора сервера. В результате проведенной обработки данных получены функции принадлежности и диапазоны изменения показателей для решения задач РСД и РВН.

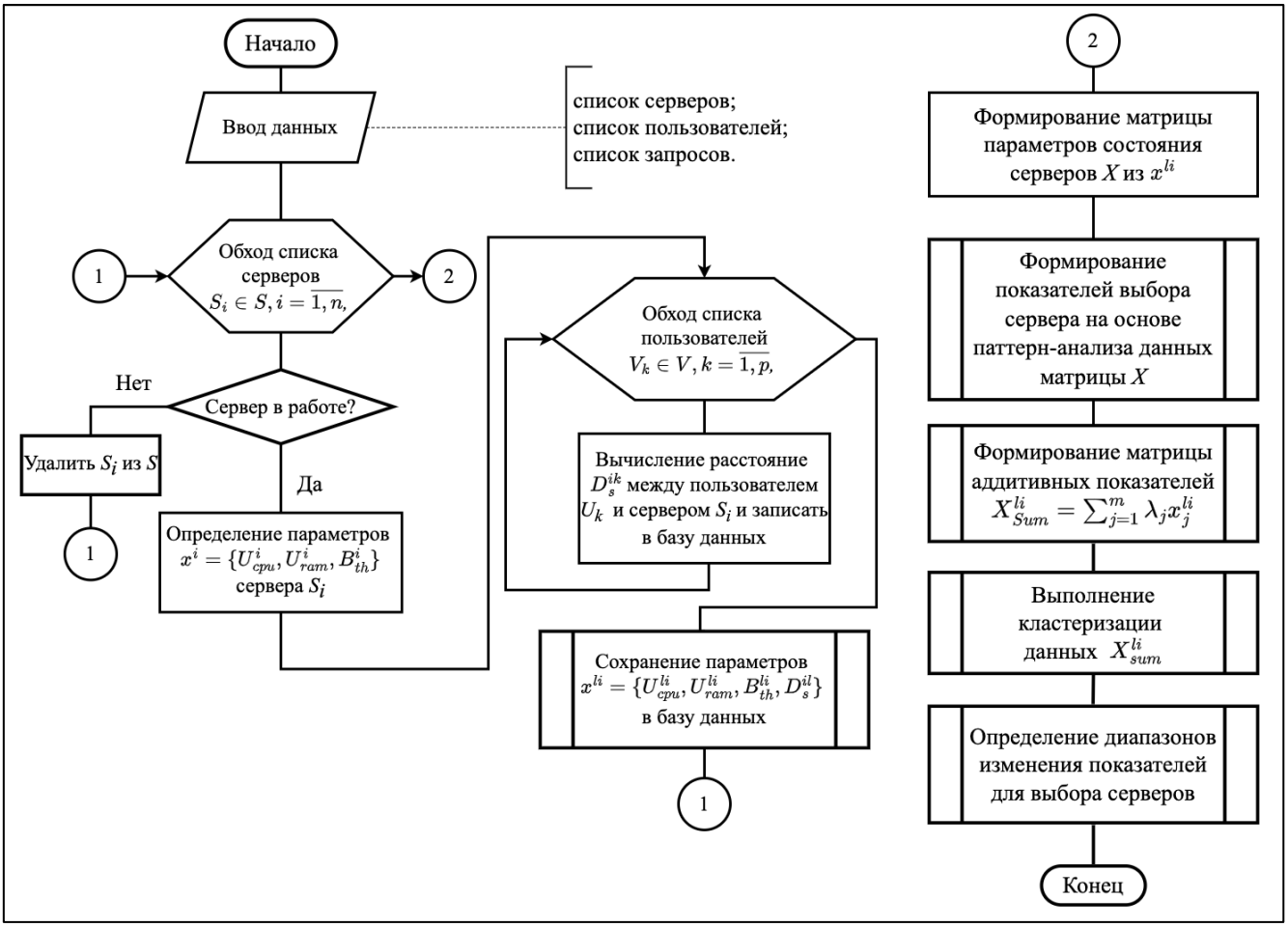

Рисунок 2 – Схема алгоритма сбора и обработки параметров состояния серверов с определением диапазонов их изменения.

Задача распределения вычислительной нагрузки ставится следующим образом: требуется отправлять запросы пользователей на выбранные сервера так, чтобы временные затраты на выполнение задач были минимальны. Тогда критерий выбора сервера (с учетом параметров его состояния) для решения задачи РВН примет вид:

$$\sum_{l=1}^{q}\sum_{i=1}^{n} \alpha^{li}\, F\left(D_s^{li}, U_{cpu}^{li}, U_{ram}^{li}\right) = \sum_{l=1}^{q}\sum_{i=1}^{n} \alpha^{li}\,(t^{li}) \rightarrow min, \qquad (1)$$

где α^{li} – признак отправки l-го запроса ($l = \overline{1,q}$, q – количество запросов), на i-ый сервер ($i = \overline{1,n}$, n – количество серверов) комплекса ($\alpha^{li} = 1$, если запрос отправлен; $\alpha^{li} = 0$ - если не отправлен); $F\left(D_s^{li}, U_{cpu}^{li}, U_{ram}^{li}\right)$ – функция, зависящая от параметров состояния серверов и характеризующая временные затраты; D_s (км) – расстояние от

клиента до сервера; U_{cpu} – загруженность центрального процессора, %; U_{ram} – загруженность оперативной памяти, %; t^{li} – время (с), затраченное i-ым сервером на выполнение l-го запроса. Ограничениями задачи (1) являются диапазоны изменения параметров состояния, записанные в следующем виде: $D_b^{min} \le D_b \le D_b^{max}$; $U_{cpu}^{min} \le U_{cpu} \le U_{cpu}^{max}$; $U_{ram}^{min} \le U_{cpu} \le U_{ram}^{max}$. Индексами *max*, *min* обозначены максимальные и минимальные значения параметров.

Задача распределения статических данных ставится следующим образом: требуется распределить запросы пользователей на загрузку данных так, чтобы минимизировать стоимостные и временные затраты на обработку данных. Тогда критерий выбора сервера для решения задачи РСД примет вид:

$$\sum_{l=1}^{q}\sum_{i=1}^{n} \alpha^{li} F\left(D_b^{li}, C_{sdr}^{li}\right) = \sum_{l=1}^{q}\sum_{i=1}^{n} \alpha^{li} \left(t^{li} + c^{li}\right) \to min, \qquad (2)$$

где α^{li} – признак отправки l-го запроса ($l = \overline{1,q}$), на i-ый сервер ($i = \overline{1,n}$), $F(D_b, C_{sdr})$ - функция, зависящая от параметров состояния серверов и характеризующая временные и стоимостные затраты; D_b – произведение расстояния от клиента до сервера D_s (км) на доступную пропускную способность B_{ch} (Кбит/с) т.е. $D_b = D_s \cdot B_{ch}$; C_{sdr} – стоимость затрат на хранение, доставку и репликацию данных (ден. ед.); t^{li} – время (с), затраченное i-ым сервером на выполнение l-го запроса. Ограничениями задачи (2) являются диапазоны изменения параметров состояния серверов, записанные в виде: $D_b^{min} \le D_b \le D_b^{max}$; $C_{sdr}^{min} \le C_{sdr} \le C_{sdr}^{max}$.

Параметры, характеризующие расстояние, загруженность ресурсов и пропускную способность канала являются переменными, в то время как стоимость постоянный параметр, так как сохраняет свое значение для каждого региона расположения вычислительного ресурса и поставщика. В связи с тем, что параметры состояния серверов имеют разные диапазоны изменения, для приведения к диапазону от 0 до 1 они нормализованы с помощью деления на максимальное значение каждого из параметров массива исходных данных согласно формуле: $\hat{x}_j^{li} = x_j^{li} / x_j^{max}$. Параметры состояния серверов представлены в виде вектора: $X^{li} = (x_1^{li}, \dots, x_j^{li} \dots, x_m^{li}), l = \overline{1,q}, i = \overline{1,n}, j = \overline{1,m}$, где q – количество запросов пользователя, n – количество серверных станций, m – количество

параметров. Обоснование выбора параметров, пригодных для решения поставленной задачи выполнено с помощью анализа паттернов (паттерн-кластеризации). Для анализа паттернов необходимо сформировать обобщенные показатели, в качестве которых использованы средние значения параметров состояния серверов, которые вычисляются по формуле: $\bar{x}_j^i = \frac{1}{l}\sum_{l=1}^{q} x_j^{li}$, $j = \overline{1,m}$, $i = \overline{1,n}$.

На *первом шаге* анализа паттернов выполняются парные сравнения смежных показателей. Параметру состояния $\bar{x}_j^i \in \bar{X}^i$ ставится в соответствие кодовая последовательность символов $r_j^i = (r_1^i, \dots, r_j^i, \dots, r_m^n)$, каждый член которой вычисляется следующим образом: $r_j^i = 1$, если $\bar{x}_j^i < \bar{x}_{j+1}^i$, $r_j^i = 0$, если $\bar{x}_j^i = \bar{x}_{j+1}^i$, $r_j^i = 0$, если $\bar{x}_j^i > \bar{x}_{j+1}^i$, где, $i = \overline{1,n}$ – номер сервера, n – количество серверов, $j = \overline{1,m-1}$– номер параметра состояния, m – количество параметров, $\bar{x}_j^i \in \bar{X}^i$.

На *втором шаге* сформированная последовательность r_j^i представляется в виде десятичного числа z^i. Таким образом исследуемые объекты (сервера) будем характеризовать вектором средних значений параметров состояния $\bar{X}^i$ и позиционным кодом $r_j^i = (r_1^i, r_2^i \dots, r_j^i, \dots, r_m^i)$, характеризующим парные отношения смежных параметров.

На *третьем шаге* выполняется кластеризация параметров путем оценки близости кодов с помощью расстояния Хемминга: $d(r_j^i, r_j^h) = \sum_{j=1}^{m-1} |r_j^i - r_j^h|$, где $i = \overline{1,n}$ – текущий номер сервера, $h = \overline{1,n}$, $i \neq h$ – следующий номер сервера n – количество серверных станций, j – номер параметра, m – количество параметров. В результате объекты относятся к кластерам по следующему правилу: если $d(r_j^i, r_j^h) = 0$, то объекты, принадлежат одному кластеру $\bar{X}^i, \bar{X}^h \in y^i$, $i = \overline{1,n}$, $h = \overline{1,n}$, $i \neq h$, в противном случае, $\bar{X}^i, \bar{X}^h$ принадлежат разным кластерам y^i, y^h.

Для получения исходных данных проведены экспериментальные исследования распределения запросов пользователей по серверам. Подготовлены URI – адреса /upload, /search для трех серверов, при обращении на которые производилась загрузка тестовых файлов и поиск ключевой последовательности символов в текстовом файле. После выполнения запросов получено 150 записей данных с показателями, D_s, U_{ram}, U_{cpu} и D_s, B_{ch}, C_{sdr}.

В ходе выполнения работы проведены расчеты для выявления паттерн-кластеров для обеих задач. Для задачи распределения вычислительной нагрузки

получено, что ни одна кодовая последовательность не повторяется, то есть метрики расстояния между паттернами не равны нулю. Таким образом, каждый из серверов относится к отдельному кластеру, тем самым, подтверждается гипотеза о том, что исходное множество параметров состояния различимо, и параметры D_s, U_{cpu}, U_{ram} пригодны для выбора сервера в задаче распределения вычислительной нагрузки.

Для задачи распределения статических данных получено, что по параметрам B_{ch}, D_s, C_{sdr}, сервера S_1 и S_2 являются структурно близкими объектами и обладают схожими значениями параметров состояния (рисунок 3в). Выдвинута гипотеза о том, что сервера S_1 и S_2 могут быть определены в один кластер. Эта гипотеза, проверена с помощью выполнения порядково-инвариантной паттерн-кластеризации и иллюстрации кусочно-линейных функции (рисунок 3) для задачи РСД. Для того, чтобы улучшить отделимость параметров серверов решено использовать произведение параметров $D_b = D_s \cdot B_{ch^h}$, характеризующих расстояние от клиента до сервера D_s и пропускную способность канала B_{ch}, т.е. в качестве одного из параметров состояния принят мультипликативный параметр D_b, который использован при решении задачи РСД. На рисунке 3г, показаны паттерны для параметров D_b, C_{sdr}.

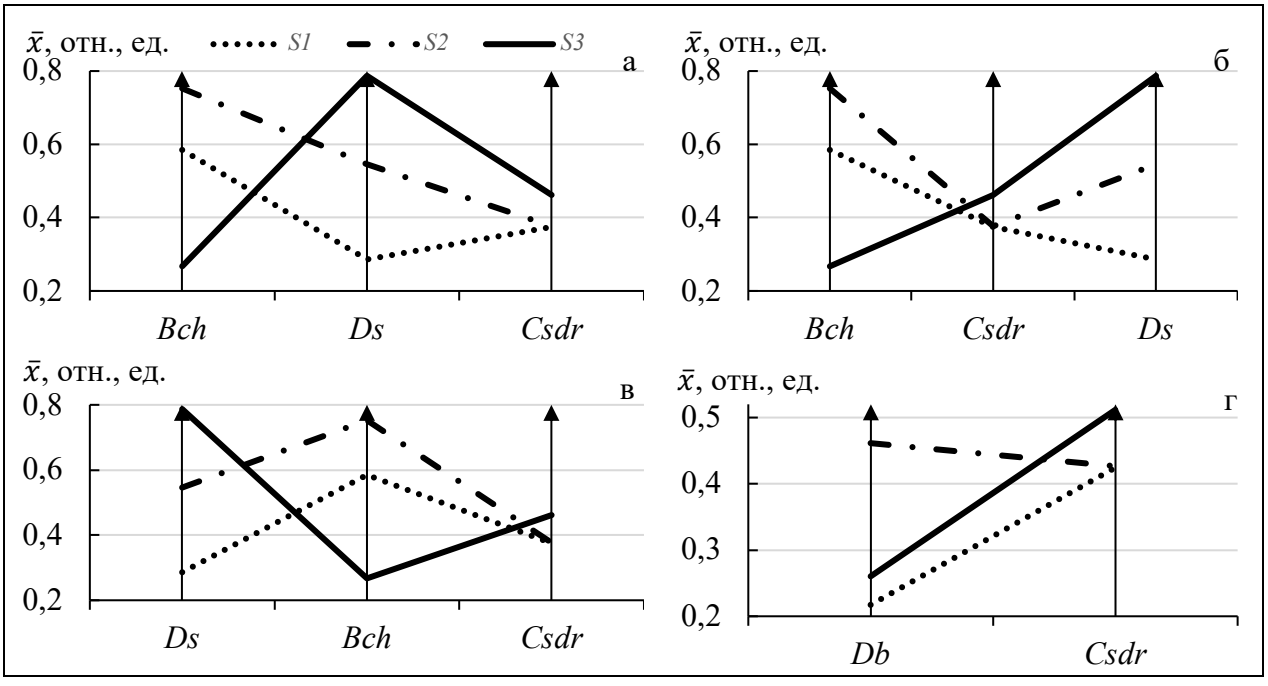

Рисунок 3 – Кусочно-линейные функции параметров на уровне средних, соответствующие паттернам: а) – $P_1 = (B_{ch}, D_s, C_{sdr})$, б) – $P_2 = (B_{ch}, C_{sdr}, D_s)$, в) – $P_3=(D_s, B_{ch}, C_{sdr})$, г) – $P_4 = (D_b, C_{sdr})$

При применении мультипликативного параметра паттерны становятся различимы. Результаты паттерн-кластеризации для паттернов $P_1 – P_3$ представлены в таблице 1.

Таблица 1 – Результаты порядково-инвариантной паттерн-кластеризации

Сервер	Кодовые последовательности и позиционные коды паттернов					
	$P_1 = (D_s, B_{ch}, C_{sdr})$		$P_2 = (B_{ch}, D_s, C_{sdr})$		$P_3 = (C_{sdr}, B_{ch}, D_s)$	
	r^i_{j1}	z	r^i_{j2}	z	r^i_{j3}	z
S_1	$r^1 = (r^1_{11}, r^1_{12}) = (1\ 2)$	**12**	$r^1 = (r^2_{11}, r^2_{12}) = (2\ 1)$	21	$r^1 = (r^1_1, r^1_2) = (2\ 2)$	*22*
S_2	$r^2 = (r^2_{21}, r^2_{22}) = (1\ 2)$	**12**	$r^2 = (r^2_{21}, r^2_{22}) = (2\ 2)$	22	$r^2 = (r^2_1, r^2_{23}) = (2\ 1)$	21
S_3	$r^3 = (r^3_{31}, r^3_{32}) = (2\ 1)$	21	$r^3 = (r^2_{31}, r^2_{32}) = (1\ 2)$	12	$r^3 = (r^3_{31}, r^3_{32}) = (1\ 1)$	11
r^i $(i = \overline{1,n})$– кодовая последовательность парных сравнений; z – позиционный десятичный код						

Для проверки выдвинутой гипотезы и дальнейшей экстракции функций принадлежности проведен кластерный анализ методом *C-средних*. В виде матрицы представлены исходные данные для каждого l-го запроса пользователя с учетом j-го параметра состояния i-го сервера.

$$X = \begin{bmatrix} \begin{bmatrix} x_1^{11} & x_2^{11} & \cdots & x_m^{11} \\ x_1^{12} & x_2^{12} & x_1^{12} & \cdots \\ \cdots & \cdots & x_j^{1i} & \cdots \\ x_1^{1n} & x_2^{1n} & \cdots & x_m^{1n} \end{bmatrix} \cdots \begin{bmatrix} x_1^{l1} & x_2^{l1} & \cdots & x_m^{l1} \\ x_1^{l2} & x_2^{l2} & x_1^{l2} & \cdots \\ \cdots & \cdots & x_j^{li} & \cdots \\ x_1^{ln} & x_2^{ln} & \cdots & x_m^{ln} \end{bmatrix} \cdots \begin{bmatrix} x_1^{q1} & x_2^{q1} & \cdots & x_m^{q1} \\ x_1^{q2} & x_2^{q2} & x_1^{q2} & \cdots \\ \cdots & \cdots & x_j^{qi} & \cdots \\ x_1^{qn} & x_2^{qn} & \cdots & x_m^{qn} \end{bmatrix} \end{bmatrix};$$

где , $i = \overline{1,n}$ – номер сервера, n – количество серверов, $j = \overline{1,m-1}$ – номер параметра состояния, m – количество параметров. Следует принять во внимание, что стандартная задача кластеризации решается, если необходимо разбить по кластерам объекты, характеризующиеся векторами своих параметров. Поэтому для выполнения процедуры кластеризации (определения нужного сервера для отправки на него задачи) для каждого l-го запроса по отношению к i-му серверу поставим в соответствие вектор, полученный суммированием параметров состояния серверов: $X^{li}_{sum} = \sum_{j=1}^{m} \lambda_j x_j^{li}, l = \overline{1,q}, i = \overline{1,n}$, где $\lambda_j \in [0,1]$ весовой коэффициент, значение которого характеризует важность параметра (при этом каждый сервер характеризуется не набором из трех показателей, а лишь одним параметром).

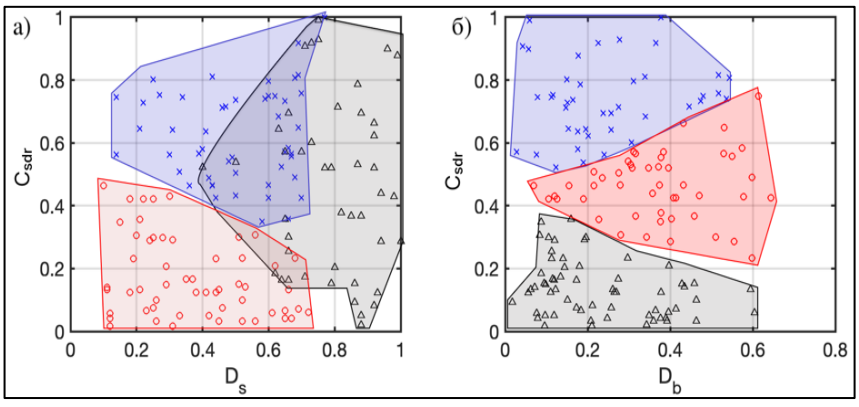

Рисунок 4 – Кластеризация запросов по параметрам: а) – C_{sdr}, D_s, б) – C_{sdr}, D_b

Результаты кластеризации тестовой выборки для параметров D_s, C_{sdr} и с мультипликативным параметром D_b^{li} показаны на рисунке 4. Кластеры на рисунке 4б не накладываются друг на друга, в отличие от рисунка 4а, что показывает возможность применения мультипликативного параметра $D_b = D_s \cdot B_{ch}$, для проведения дальнейшей процедуры выбора сервера. На основе кластерного анализа для каждого из запросов определены степени принадлежности к кластерам S_1-S_3 (по расстояниям от центров кластеров), по которым сформированы обобщенные функции принадлежности к лингвистическим термам для каждой входной переменной нечеткого логического вывода (для переменной D_s на рисунке 5).

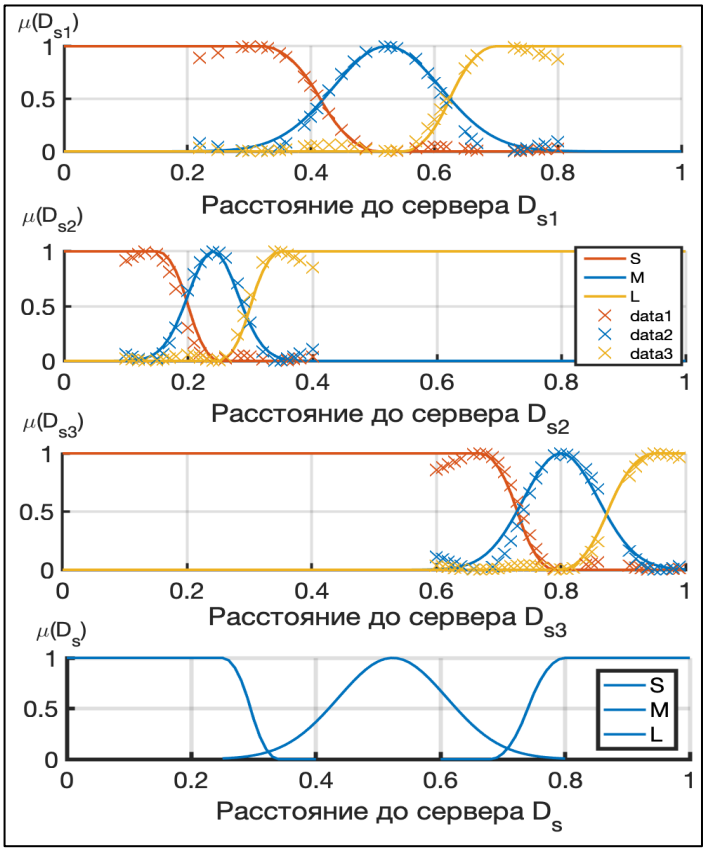

Рисунок 5 – Формирование обобщенных функций принадлежности к термам входной переменной D_s

Аналогичные функции принадлежности получены для каждого из параметров состояния серверов для обеих задач. Полученные диапазоны изменения параметров серверов использованы в качестве исходных для аналитико-имитационной модели. Отметим, что осуществление выбора сервера при помощи кластеризации параметров состояния серверов является ресурсно-затратным, поэтому в работе использован метод выбора сервера с помощью аппарата нечеткой логики.

В третьей главе представлен аналитико-имитационный метод балансировки нагрузки, включающий получение аналитических зависимостей (в результате обработки данных о состоянии вычислительных ресурсов) и модельные исследования работы сервера-балансира (основанного на нечетком логическом выводе) для РСД и РВН. На рисунке 7 приведена схема модели распределения нагрузки по вычислительным ресурсам. Имитационная модель серверного комплекса позволяет определять его характеристики и изменения состояния во времени при заданных потоках заявок, поступающих на входы системы, и обосновать алгоритм выбора сервера.

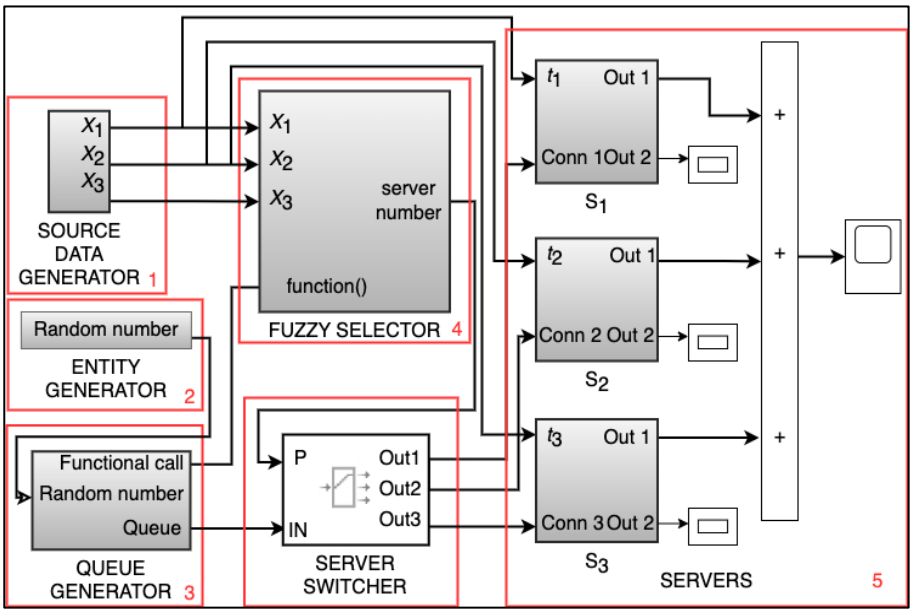

Рисунок 7 – Схема модели серверного комплекса для задачи РСД

В модели серверного комплекса выделены подсистемы (1-5), выполняющие основные функции обработки запросов: генерация параметров состояния серверов (1) и потока заявок (2), создание очереди запросов (3). Подсистема нечеткого селектора Fuzzy Selector (4) выполняет выбор сервера на основе нечеткого логического вывода (рисунок 8а), подсистема Servers (5) обрабатывает запросы на выбранном сервере. Переключатель серверов (Server switcher) перенаправляет запросы на выбранный сервер. Подсистема Servers представляет каналы обслуживания (сервера), которые принимают на вход запросы и задерживают их в течение времени обслуживания (рисунок 8б).

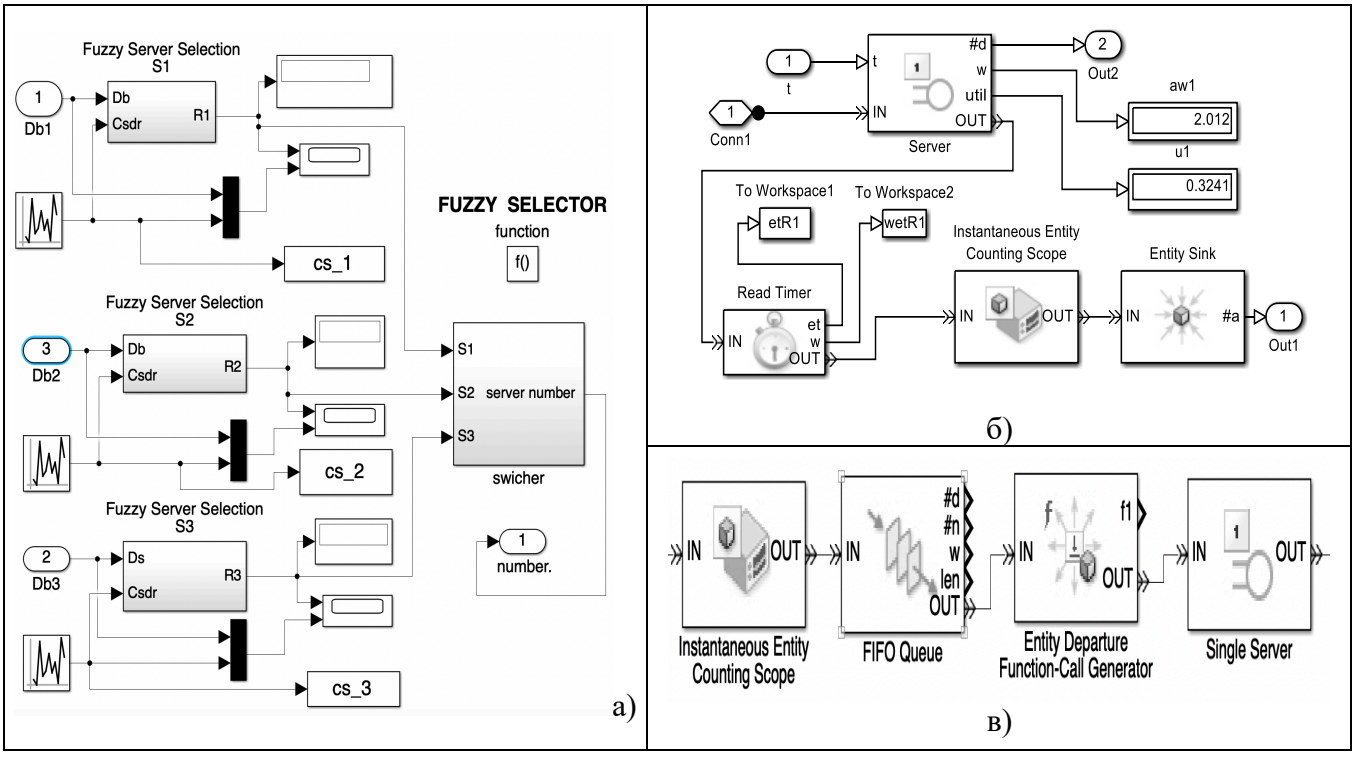

Рисунок 8 - Модель выбора сервера: а) – подсистема нечеткого выбора сервера; б) – подсистема серверов; в) – подсистема генерации очереди.

На рисунке 8в представлена подсистема генерации очереди запросов. В качестве входной информации для принятия решения о выборе сервера используется вектор входных параметров состояние серверов X^{li}. Выходной параметр обозначен как R – решение о выборе сервера, которое может иметь значения: P – позитивное; Z – нейтральное; N – негативное. Операция обратного преобразования нечетких переменных в четкие (дефаззификация) формирует значения выходной переменной. При этом четкий вывод о пригодности каждого сервера для выбора осуществляется путем нахождения взвешенного среднего: $R = \sum_{j=1}^{m} \mu(R_j) R_j / \sum_{j=1}^{m} \mu(R_j)$, где R – четкое значение выходной переменной; R_j – значение выходной переменной для j-го терма с единичным значением степени принадлежности; $\mu\ (R_j)$ – степень принадлежности к j-му терму; m=3 – число термов; R принимает значения: N=0; Z=1; P=2.

Для задачи РВН получено, что на основе метода с нечетким распределением серверным комплексом обработано 56 запросов: тогда как круговым методом обработано 53 запроса (рисунок 9). Длительности пребывания запросов в системе в случае учета показателей работы серверов существенно уменьшились (не более 18 с, а при круговом распределении превышают 24 с). Среднее время пребывания запросов в системе сократилось на 20-60% для среднего времени между поступлением запросов t_{ig} от 1 с до 1.8 с. Для задачи распределения данных получены стоимостные (рисунок 10а) и временные (рисунок 10б) характеристики.

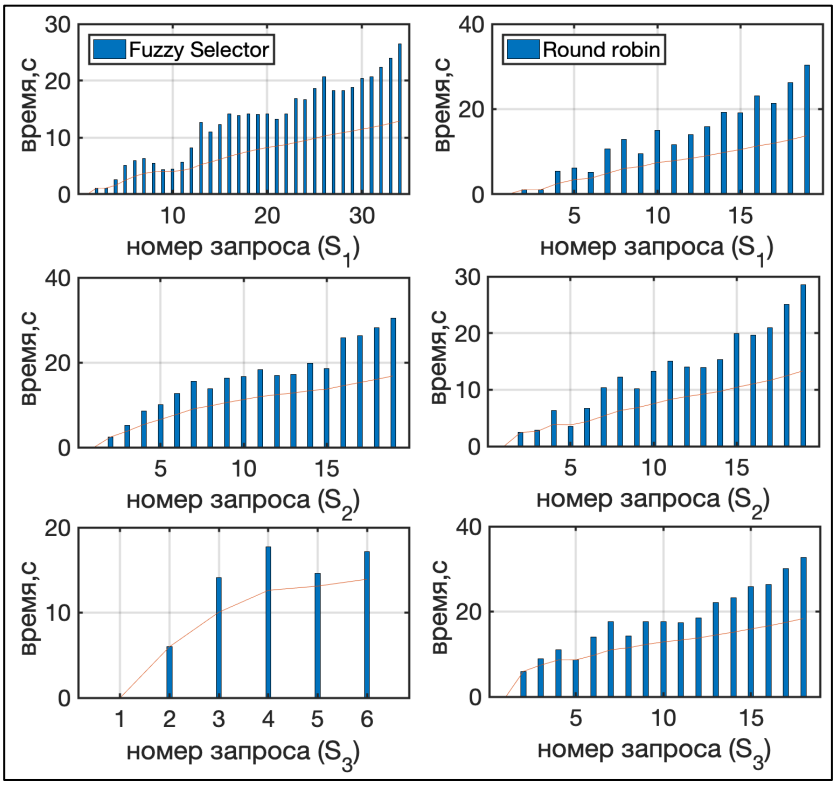

Рисунок 9. Результаты распределения заявок по серверам: длительность обработки заявок в системе.

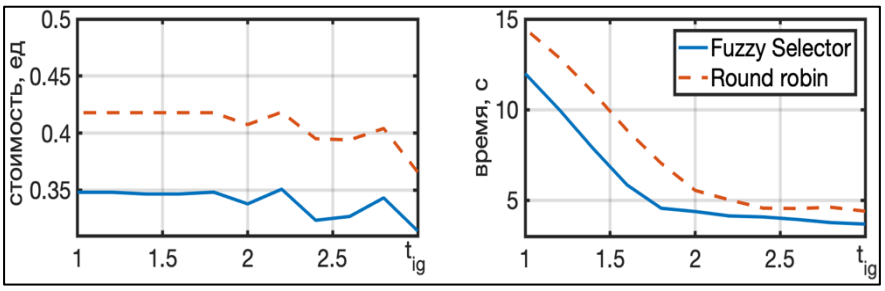

Рисунок 10 – Зависимость средних значений времени (а) и стоимости (б) обработки запросов от длительности паузы между запросами

Средняя стоимость обработки запросов уменьшилась на 10-17% по сравнению с круговым методом распределения (рисунок 10а). Одновременно с уменьшением стоимости обработки запросов снизилось и время, затраченное на обработку запросов. Получено, что среднее время обработки запросов уменьшилось на величину от 5 до 20% (рисунок 10б). Отметим, что при длительности паузы между поступлением запросов t_{ig} от 1 с до 1,6 с. изменение стоимости перестает происходить, так как система не успевает обработать большего количества запросов. На рисунке 11 представлено распределение запросов по серверам для t_{ig}=1,6. В случае использования метода кругового распределения нагрузки Round Robin запросы распределены по серверам равномерно, но при этом одинаково загружаются как высоко нагруженные серверы с высокой стоимостью, так и серверы с низкой нагрузкой и низкой стоимостью. Тогда как нечеткий метод распределяет больше запросов на менее загруженные и более близкие к пользователю сервера с низкой и средней стоимостью, обеспечивая более высокую скорость обработки запросов пользователя. Таким образом, распределение данных с помощью нечеткого метода улучшает временные и стоимостные характеристики обслуживания запросов серверным комплексом, тем самым минимизируя целевые функции (1) и (2).

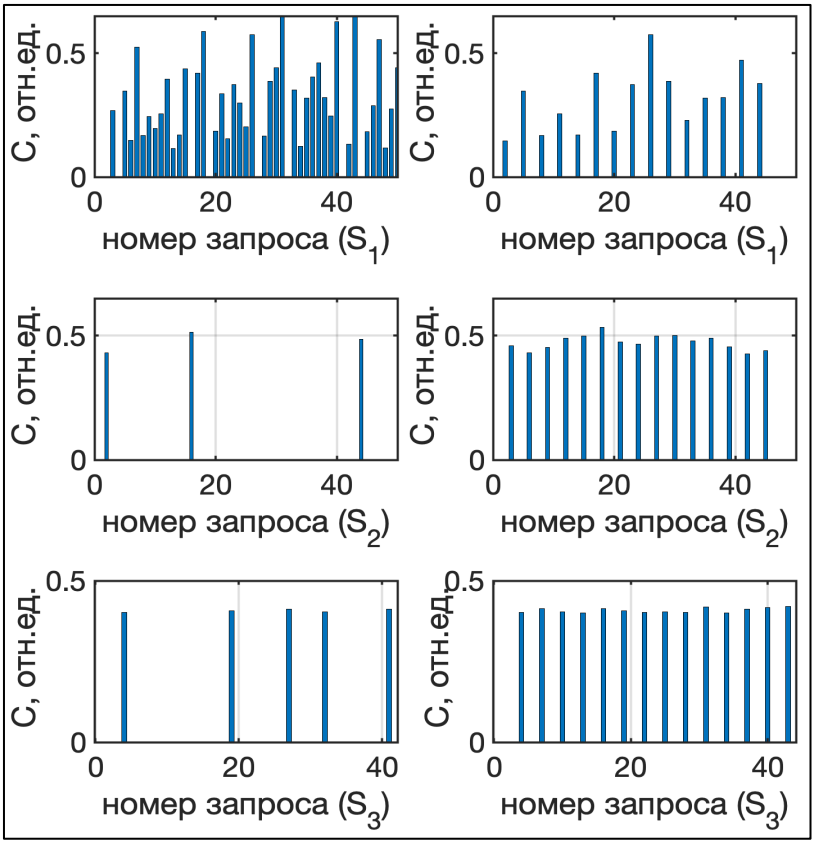

Рисунок 11 – Результаты распределения заявок по серверам: стоимость обработки заявок в системе.

В **четвертой главе** описаны результаты исследований распределения вычислительной нагрузки по ресурсам облачного серверного кластера. Представлен алгоритм параллельных вычислений для распределения нагрузки и структура программного комплекса для проведения экспериментальных исследований. На основе схемы клиент-серверного приложения (рисунок 1) и с учетом результатов экспериментальных исследований разработана система балансировки нагрузки (рисунок 12). Для того чтобы процедура выбора подходящего сервера не влияла на время выполнения запроса пользователя принято решение о переносе вычислений по выбору сервера в параллельный процесс (рисунок 13).

Рисунок – 12 Схема системы балансировки

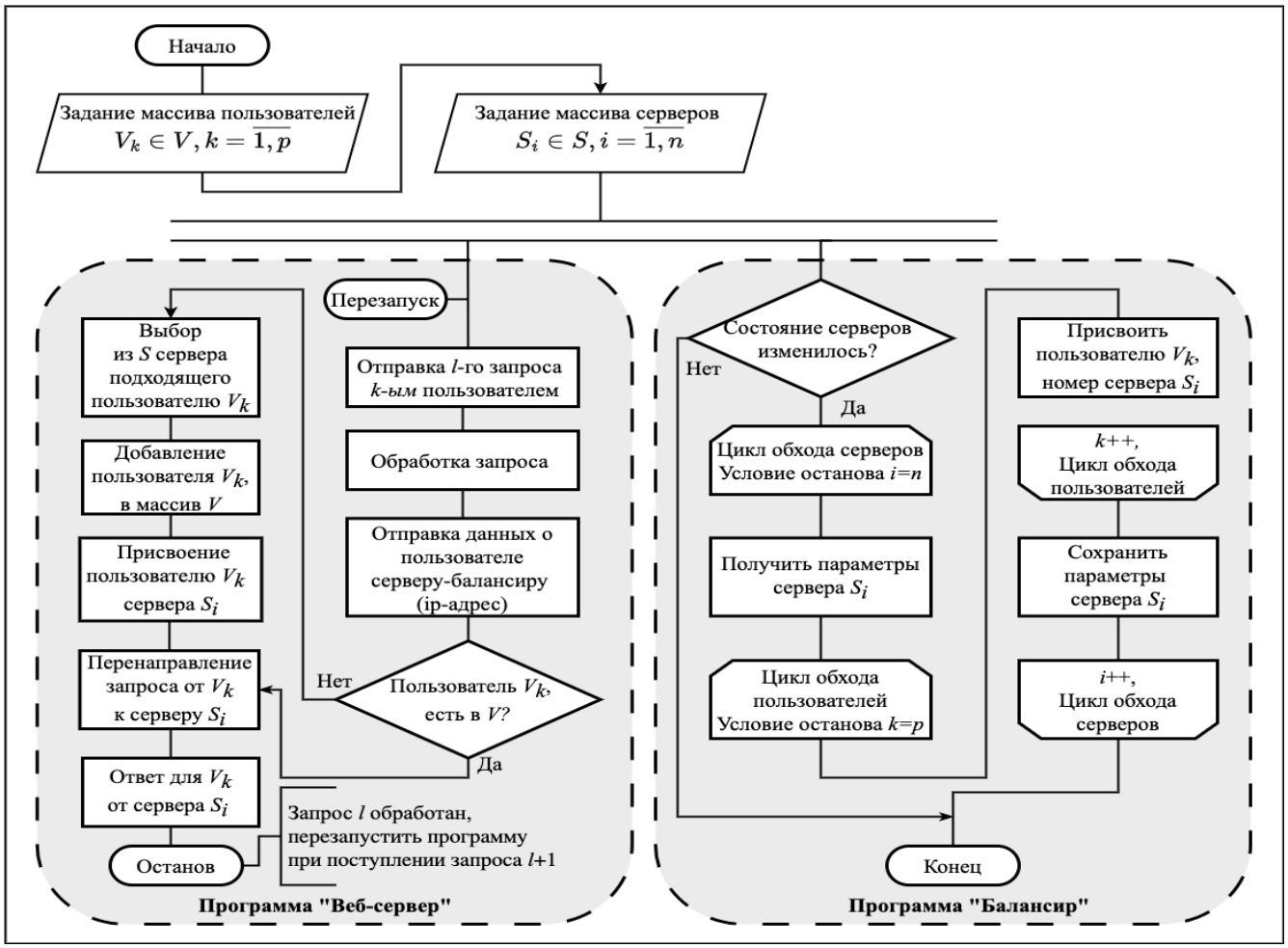

Рисунок – 13 Алгоритм балансировки нагрузки с параллельными вычислениями

Таким образом, серверное приложение разделено на три части: веб-сервер (программа «Веб-сервер»), состоящий из модуля обработки запроса и модуля передачи данных; вычислительных серверов (с программой «Агент»), решающих поставленную задачу, и сервера параллельных вычислений (программа «Балансир»), который осуществляет непосредственный выбор сервера для обработки запроса.

Для того, чтобы оценить время обработки запросов серверным приложением было проведено три натурных эксперимента E_1–E_3. Каждый эксперимент проводился с применением кругового распределения, кластеризации, нечеткой логики и алгоритма балансировки, предоставляемого провайдером (Google Cloud Platform Balancer). В эксперименте E_1 на сервера не оказывалось дополнительной нагрузки. В эксперименте E_2 сервера S_2, S_3 были нагружены на 50% от имеющейся мощности (по параметрам U_{cpu}, U_{ram}). В эксперименте E_3 нагрузка составила 90% от мощности аппаратного ресурса. Результаты экспериментов приведены на рисунке 14.

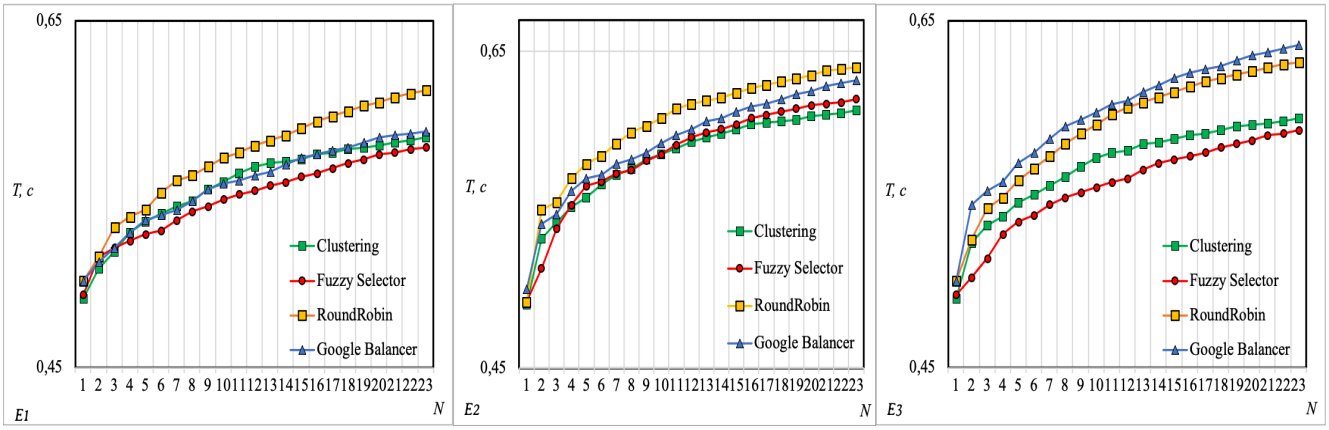

Рисунок 14 – Эксперименты распределения вычислительной нагрузки E_1–E_3

Получено, что разработанный метод и алгоритм распределения нагрузки, увеличивают скорость обработки запросов пользователей до 10% по сравнению с круговым распределением и до 7% по сравнению с методами балансировки провайдера. В таблице 2 отмечены лучшие временные результаты обработки запросов тестовой выборки, свидетельствующие о преимуществе предлагаемого метода.

Таблица 2 – Результаты экспериментов E_1–E_3

Эксперимент	Метод распределения нагрузки			
	Кластеризация (*C-means*)	Нечеткий селектор (*Fuzzy Selector*)	Круговое распределение (*Round Robin*)	Балансир провайдера (*Google Balancer*)
	T_s – среднее время обработки запросов тестовой выборки (с)			
E_1	3402	3320	3434,3	3466,7
E_2	3424	3426	3565	3525
E_3	3440	3345	3546	3571

ЗАКЛЮЧЕНИЕ

В результате проведенных исследований получены новые научные и практические результаты, направленные на повышение быстродействия и производительности работы клиент-серверных информационных систем.

1. В результате анализа работ, посвященных вопросам балансировки нагрузки в серверном комплексе выявлено, что известные методы распределения нагрузки не учитывают или учитывают не в полной мере состояние вычислительных ресурсов (учитываются данные о доступности ресурса, а не о его состоянии). Установлено, что для увеличения количества обработанных запросов и уменьшение временных затрат на обработку необходимо распределять нагрузку в зависимости от состояния вычислительных ресурсов.

2. Выполнено обоснование показателей для выбора сервера на основе паттерн-кластеризации параметров состояния серверного комплекса. Показано, что для оценки качества балансировки вычислительной нагрузки пригодны методы как порядково-фиксированной (порядково-зависимой), так и порядково-инвариантной (порядково-независимой) паттерн-кластеризации. Для задачи РСД применение порядково-инвариантной кластеризации позволило выявить необходимость привлечения мультипликативного показателя (произведения параметров, характеризующих расстояние от клиента до сервера и пропускную способность канала) для лучшей отделимости данных, используемых при выборе сервера, что в свою очередь обеспечило повышение быстродействия при доставке данных пользователю.

3. Показано, что для решения задачи балансировки нагрузки пригодны методы кластерного анализа данных о состоянии серверов, применение которых позволило разделить запросы пользователей по серверам. На основании выявленных показателей качества балансировки и нечеткой кластеризации данных о состоянии серверов получены новые результаты, которые представляют собой сформированные продукционные правила для выбора сервера с применением нечетного логического вывода. Установлены численные границы диапазонов изменения параметров состояния вычислительных ресурсов и определены степени их принадлежности к сформированным лингвистическим термам.

4. Разработан аналитико-имитационный метод распределения нагрузки на основе нечеткого логического вывода (положенного в основу работы сервера-балансира), включающий аналитическую обработку экспериментальных данных о

состоянии вычислительных ресурсов в совокупности с модельными исследованиями, позволяющий выбрать и обосновать параметры алгоритма балансировки, обеспечивая повышение быстродействия и отказоустойчивости высоконагруженной клиент-серверной информационной системы.

Установлено, что применение предлагаемого метода позволяет значительно улучшить качество работы системы балансировки нагрузки (по сравнению с традиционно используемым круговым распределением). Так, для задачи РСД получено, что средняя стоимость обработки запросов уменьшилась на 10-17%, а количество обработанных запросов увеличилось на 5-20%. Для задачи РВН получено, что среднее время пребывания запросов сократилось на 20-60%, а количество обработанных запросов увеличилось на 20-60%.

5. Установлено, что при распределении вычислительной нагрузки в серверном комплексе на основе алгоритма параллельных вычислений существенно увеличилось быстродействие системы, что позволило выполнить больший объем вычислений и провести эксперименты с облачными ресурсами серверного кластера. Получено, что предлагаемая структура программного комплекса, включающая параллельно работающие программы балансировки и принятия решений о выборе сервера (на основе нечеткого логического вывода), позволяет повысить скорость обработки запросов пользователя клиент-серверной информационной системы. Проведенные экспериментальные исследования распределения нагрузки в облачном кластере серверов показали, что скорость обработки запросов пользователей увеличилась на 7% по сравнению с результатами работы балансировщика, предоставляемого провайдером вычислительных ресурсов.

ОСНОВНЫЕ РАБОТЫ, ОПУБЛИКОВАННЫЕ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в научных журналах, рекомендованных ВАК при Минобрнауки России

1. **Викулов, Е. О.** Исследование распределения данных высоконагруженных веб-приложений с применением нейросетевых технологий / Е. О. Викулов // Омский научный вестник. – 2018. – № 6 (162). – С. 244–246.

2. **Викулов, Е.О.** Имитационное моделирование распределения вычислительной нагрузки между серверными станциями с использованием нечеткого логического вывода / Е.О. Викулов, О.В. Денисов, В.А. Мещеряков, Л.А. Денисова, // Автоматизация в промышленности. 2021. – №9. – С.7-14.

3. **Викулов, Е.О.** Распределение вычислительной нагрузки в серверном комплексе с помощью деревьев решений / Е.О. Викулов // Известия Тульского государственного университета. Технические науки. – 2023. – №12. – С. 457-461

4. **Викулов, Е.О.** Распределение данных по серверам на основе нечеткого логического вывода / Е.О. Викулов, Л.А. Денисова // Известия Тульского государственного университета. Технические науки. – 2024. – №3. – С. 4-8

Статьи в изданиях, индексируемых в базе Scopus

5. **Vikulov, E.** Event-Driven Simulation of Server Stations Load Balancing / E. Vikulov, O. Denisov, V. Meshcheryakov // International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM) / Institute of Electrical and Electronics Engineers Inc. – Moscow, 2018. – P. 8728849.

6. **Vikulov, E.O.** Data distribution system: preparation of server stations data / E. O. Vikulov, O. V. Denisov, L. A. Denisova // Journal of Physics: Conference Series. – 2018. – Vol. 1050. – P. 012097.

7. **Vikulov, E.O.** Data distribution system: clustering based on neural network technologies / E. O. Vikulov, L. A. Denisova // IOP Conference Series Materials Science and Engineering. – 2019. – Vol. 537, no. 5. – P. 052030.

8. **Vikulov, E.O.** Simulation of data distribution between server stations using fuzzy technologies / E. O. Vikulov, O. V. Denisov, V. A. Meshcheryakov L. A. Denisova // Journal of Physics: Conference Series. Electronic collection. – 2020. – C.012175.

9. Denisov, O.V. Load balancing in data distribution systems / O. V. Denisov, **E. O. Vikulov** // Journal of Physics: Conference Series. IV International Scientific and Technical Conference "Mechanical Science and Technology Update", MSTU 2020. – C. 012003.

Статьи в других рецензируемых научных журналах

10. **Викулов, Е.О**. Автоматизированное распределение больших объемов данных высоконагруженных систем / Е. О. Викулов, Е. А. Леонов, Л. А. Денисова // Динамика систем, механизмов и машин. – 2014. – № 3. – С. 146–149.

11. **Викулов, Е.О.** Применение методов кластеризации при распределении больших объёмов данных высоконагруженных систем / Е.О. Викулов, Е.А. Леонов // Информационные технологии и автоматизация управления. – ОмГТУ – Омск, 2014. – С. 7-12.

12. Денисов, О.В. Распределение данных в информационной системе с помощью сервера балансира / Денисов О.В., **Викулов Е.О.** // Прикладная математика и фундаментальная информатика. – 2019. Т. 6. № 4. – С. 46-57.

Статьи в материалах конференций

13. **Викулов, Е.О.** Распределение данных и вычислений высоконагруженных веб-приложений / Е.О. Викулов, Е.А. Леонов // Информационные технологии и автоматизация управления. – ОмГТУ – Омск, 2013. – С. 141–144.

14. **Викулов, Е.О.** Распределение данных высоконагруженных веб-приложений / Е. О. Викулов // Информационные технологии и автоматизация управления. – ОмГТУ – Омск, 2016. – С. 73–78.

15. **Викулов, Е.О.** Моделирование распределения нагрузки между серверными станциями / Е.О. Викулов, О.В. Денисов, М.А. Рудгальский // Информационные технологии и автоматизация управления. – ОмГТУ – Омск, 2016. – С. 38–43.

16. **Викулов, Е.О.** Исследование влияния системы доменных имен на скорость доставки статических данных / Е.О. Викулов // Информационные технологии и автоматизация управления. – ОмГТУ – Омск, 2016. – С. 91–96.

17. **Викулов, Е.О.** Анализ данных о состоянии серверных станций высоконагруженных веб-приложений / Е.О. Викулов // Информационные технологии и автоматизация управления. – ОмГТУ – Омск, 2019. – С. 62-67.

Государственная регистрация программ для ЭВМ

18. Свид. о гос. рег. прогр. для ЭВМ № 2019661730 Российская Федерация. Программный комплекс сбора данных о состоянии серверных станций № 2021617727; заяв. 25.05.2021; опубл. 02.06.2021 / **Е.О. Викулов**

19. Свид. о гос. рег. прогр. для ЭВМ № 2022662254 Российская Федерация. Программный комплекс параллельных вычислений распределения нагрузки облачных серверных станций № 2022660916; заяв. 14.06.2022; опубл. 30.06.2022 / **Е.О. Викулов**

Печатается в авторской редакции
Подписано в печать 24.04.2024.
Формат 60×84/16.Тираж 100 экз.

Типография: 644050, г. Омск, пр. Мира д. 11, т. 65-32-08
Омский государственный технический университет, научное издательство