На правах рукописи

СИДОРЕНКО Евгений Васильевич

УПРАВЛЕНИЕ ПРОЦЕССАМИ МОНИТОРИНГА ПРОИЗВОДИТЕЛЬНОСТИ КОЛЛЕКТИВОВ ВСТРАИВАЕМЫХ ОБЪЕКТОВ БОЛЬШИХ ПРОГРАММНЫХ СИСТЕМ В ОБЛАЧНЫХ АРХИТЕКТУРАХ

Специальность: 2.3.5. Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Воронеж – 2025

Работа выполнена в ФГБОУ ВО «Воронежский государственный технический университет».

Научный руководитель: **Рындин Никита Александрович**, доктор технических наук, доцент

Официальные оппоненты: **Таныгин Максим Олегович,** доктор технических наук, доцент, ФГБОУ ВО «Юго-Западный государственный университет» (г. Курск), декан факультета фундаментальной и прикладной информатики, профессор кафедры информационной безопасности

Затонский Андрей Владимирович, доктор технических наук, профессор, ФГАОУ ВО «Пермский национальный исследовательский политехнический университет», Березниковский филиал, заведующий кафедрой «Автоматизация технологических процессов»

Ведущая организация: **ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина»**

Защита состоится «27» июня 2025 года в 12^{00} часов в конференц-зале на заседании диссертационного совета 24.2.286.04, созданного на базе ФГБОУ ВО «Воронежский государственный технический университет», по адресу: г. Воронеж, Московский просп., д. 14, ауд. 216.

С диссертацией можно ознакомиться в научно-технической библиотеке ФГБОУ ВО «Воронежский государственный технический университет» и на сайте https://cchgeu.ru.

Автореферат разослан «12» мая 2025 г.

Ученый секретарь диссертационного совета  Гусев Константин Юрьевич

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. С быстрым развитием датчиков и информационных технологий, обеспечивающих промышленное производство, сетевые данные, которые представляют взаимодействия между взаимосвязанными объектами, представлены в коллективах встраиваемых объектов больших программных систем.

Статистический мониторинг процессов служит эффективным инструментом поддержки точного и своевременного принятия решений в. Применение подходов к мониторингу статистических процессов для мониторинга сетей значительно облегчает раннее обнаружение потенциальных сбоев в сложных реляционных системах и, следовательно, в последние годы все чаще изучается. Выбор эффективного метода сетевого мониторинга основывается на оценке эффективности методов-кандидатов. Однако исследований по систематической оценке и сравнению методов сетевого мониторинга очень мало. В частности, возможность частого сбора данных с помощью современных измерительных устройств, как правило, приводит к возникновению автокорреляций между сетями. Тем не менее, методов оценки производительности для автокоррелированных сетей крайне не хватает. Большой вклад в развитие методов статистического мониторинга программных систем внесли Афанасьев В.Н., Домрачев В.Г., Городецкий А.Я., Скуратов А.К., Farahani E.M., He Z., Kubacki M. Разумной идеей кажется разработать модель статистического мониторинга процессов с большими данными в компьютерных сетях, учитывающую свойства автокорреляции процессов образования и распада части локальных структур.

Сети обычно используются для описания взаимосвязей между различными объектами в сложных системах. Контрольные диаграммы как правило, используются для мониторинга того, остается ли процесс со случайными шумами статистически контролируемым с течением времени. Ожидается, что хорошая контрольная диаграмма быстро обнаружит изменения в процессе в случае выхода из-под контроля и не вызовет ложной тревоги, когда процесс находится под контролем. Однако эти две цели не могут быть достигнуты одновременно. Разрешение этого противоречия является значимой задачей.

Мониторинг производительности и обнаружение аномалий являются основными целями при проектировании и обслуживании электронных устройств и систем. В последние годы они усложняются из-за усложнения аппаратного и программного обеспечения. Следовательно, важным моментом является сбор репрезентативных выборок сигналов и выявление характерных особенностей, позволяющих оценить рабочие характеристики устройств. Это приводит к необходимости эффективного анализа временных рядов. Анализ рабочих характеристик различных физических устройств или систем предполагает измерение различных параметров (метрик) во времени. Собранные выборки данных представляют собой временные ряды, которые необходимо исследовать для получения характерных признаков, коррелирующих с исследуемыми свойствами. Отсюда понятна необходимость создания архитектуры программного обеспечения оптимизации облачных сервисов «программное

обеспечение как услуга» для решения задач мониторинга.

Таким образом, актуальность темы диссертационного исследования продиктована необходимостью дальнейшего развития средств управления процессами мониторинга производительности коллективов встраиваемых объектов больших программных систем в облачных архитектурах на основе учета автокорреляции образования и распада части локальных структур.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный технический университет» «Вычислительные комплексы и проблемно-ориентированные системы управления».

Целью работы является разработка моделей и алгоритмов управления процессами мониторинга производительности коллективов встраиваемых объектов больших программных систем в облачных архитектурах на основе учета автокорреляции образования и распада части локальных структур.

Задачи исследования. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ проблем создания моделей и алгоритмов управления процессами мониторинга производительности коллективов встраиваемых объектов больших программных систем в облачных архитектурах на основе учета автокорреляции образования и распада части локальных структур.

2. Разработать модель статистического мониторинга процессов с большими данными в компьютерных сетях, учитывающую свойства автокорреляции процессов образования и распада части локальных структур на основе контрольных диаграмм.

3. Модифицировать алгоритмы мониторинга производительности коллективов встраиваемых объектов больших программных систем для обеспечения возможности иерархического анализа данных и улучшения интерпретируемости результатов.

4. Создать оптимизационную модель и алгоритмы планирования пропускной способности облачных сервисов SaaS для нахождения компромисса между стоимостью ресурсов и штрафом за задержку выполнения во встраиваемых облачных приложениях "программное обеспечение как услуга" с точки зрения поставщика облачных услуг.

5. Разработать динамический алгоритм определения минимальной верхней границы нулевого штрафа, обеспечивающий получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф.

6. Разработать структуру программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга, обеспечивающую принятие решений о допустимости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания.

Объект исследования: процессы мониторинга производительности коллективов встраиваемых объектов больших программных систем.

Предмет исследования: средства управления процессами мониторинга производительности коллективов встраиваемых объектов больших

программных систем в облачных архитектурах на основе учета автокорреляции образования и распада части локальных структур.

Методы исследования. При решении поставленных в диссертации задач использовались методы теории графов, теории вероятностей, теории принятия решений, а также методы объектно-ориентированного программирования.

Тематика работы соответствует следующим пунктам паспорта специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: п.3 «Модели, методы, архитектуры, алгоритмы, языки и программные инструменты организации взаимодействия программ и программных систем»; п.9 «Модели, методы, алгоритмы, облачные технологии и программная инфраструктура организации глобально распределенной обработки данных».

Научная новизна работы. В диссертации получены следующие результаты, характеризующиеся научной новизной:

1. Модель статистического мониторинга процессов с большими данными в компьютерных сетях, отличающаяся учетом свойств автокорреляции процессов образования и распада части локальных структур на основе контрольных диаграмм и обеспечивающая моделирование сдвигов относительно уровней плотности, взаимности, изменчивости степени и свойств транзитивности сети.

2. Алгоритмы мониторинга производительности коллективов встраиваемых объектов больших программных систем, отличающиеся предварительной кластеризацией и идентификацией поведенческих последовательностей, удовлетворяющих условиям сходства, обеспечивающие возможность иерархического анализа данных и улучшающие интерпретируемость результатов.

3. Оптимизационная модель и алгоритмы планирования пропускной способности облачных сервисов SaaS, отличающиеся форсированным решением NP-сложной задачи минимизации затрат на задержку выполнения запросов и на ресурсы развернутых экземпляров и обеспечивающие нахождение компромисса между стоимостью ресурсов и штрафом за задержку выполнения во встраиваемых облачных приложениях "программное обеспечение как услуга" с точки зрения поставщика облачных услуг.

4. Динамический алгоритм определения минимальной верхней границы нулевого штрафа, отличающийся использованием ожидающей очереди неисполненных запросов и обеспечивающий получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф.

5. Структура программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга, отличающаяся интеграцией системы мониторинга и сервисов SaaS и обеспечивающая принятие решений о допустимости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания.

Теоретическая и практическая значимость исследования заключается в разработке специальных средств математического и программного

обеспечения управления процессами мониторинга производительности коллективов встраиваемых объектов больших программных систем в облачных архитектурах на основе учета автокорреляции образования и распада части локальных структур.

Теоретические результаты работы могут быть использованы в проектных и научно-исследовательских организациях, занимающихся разработкой методов управления процессами мониторинга распределенного программного обеспечения.

Положения, выносимые на защиту

1. Модель статистического мониторинга процессов с большими данными в компьютерных сетях учитывает свойства автокорреляции процессов образования и распада части локальных структур на основе контрольных диаграмм.

2. Модифицированные алгоритмы мониторинга производительности коллективов встраиваемых объектов больших программных систем обеспечивают возможность иерархического анализа данных и улучшения интерпретируемости результатов.

3. Оптимизационная модель и алгоритмы планирования пропускной способности облачных сервисов SaaS решают задачу поиска компромисса между стоимостью ресурсов и штрафом за задержку выполнения во встраиваемых облачных приложениях "программное обеспечение как услуга" с точки зрения поставщика облачных услуг.

4. Динамический алгоритм определения минимальной верхней границы нулевого штрафа обеспечивает получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф.

5. Структура программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга обеспечивает принятие решений о допустимости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания.

Результаты внедрения. Основные результаты внедрены в ООО «Центр информационных технологий» (г. Воронеж) при проектировании распределенной информационно-вычислительной системы управления мониторингом программных систем, в учебный процесс Воронежского государственного технического университета в рамках дисциплин: «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: XXIX International Open Science Conference «Modern informatization problems in the technological and telecommunication systems analysis and synthesis» (Yelm, WA, USA, 2024); VII Международной НПК «Наука и технологии: перспективы развития и применения» (Петрозаводск, 2024); XXX International Open Science Conference «Modern informatization problems in simulation and social technologies» (Yelm, WA, USA, 2025), а также на научных семинарах кафедр САПРИС и искусственного интеллекта и цифровых технологий ВГТУ (2019-2025 гг.).

Достоверность результатов обусловлена корректным использованием теоретических методов исследования и подтверждена результатами сравнительного анализа данных вычислительных и натурных экспериментов.

Публикации. По результатам диссертационного исследования опубликовано 13 научных работ (4 – без соавторов), в том числе 7 – в изданиях, рекомендованных ВАК РФ (из них 2 – в изданиях, индексируемых в Scopus и WoS и одно свидетельство о регистрации программы для ЭВМ). В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [4] - модель статистического мониторинга процессов с большими данными в компьютерных сетях, учитывающую свойства автокорреляции процессов образования и распада части локальных структур; [2, 6] - алгоритмы мониторинга производительности коллективов встраиваемых объектов больших программных систем для обеспечения возможности иерархического анализа данных и улучшения интерпретируемости результатов; [1, 5, 12] - оптимизационная модель и алгоритмы планирования пропускной способности облачных сервисов SaaS для нахождения компромисса между стоимостью ресурсов и штрафом за задержку выполнения; [13] - динамический алгоритм определения минимальной верхней границы нулевого штрафа, обеспечивающий получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф; [6, 7, 8] - структура программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга, обеспечивающего принятие решений о допустимости конфигурации сервиса с точки зрения штрафов.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 180 наименований. Работа изложена на 188 страницах основного текста.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность исследования, сформулированы его цель и задачи, научная новизна и практическая значимость полученных результатов, приведены сведения об апробации и внедрении работы.

В первой главе исследуются проблемы управления процессами мониторинга производительности коллективов встраиваемых объектов больших программных систем в облачных архитектурах. Отмечено, что повысить эффективность управления процессами мониторинга можно путем разработки статистического мониторинга стохастических процессов на основе оценки производительности для автокоррелированных сетей, постановки оптимизационной задачи использования ограниченных ресурсов для достижения высокой производительности, а также ограниченного энергопотребления для встраиваемых систем, алгоритмизации управления эластичностью для планирования мониторинговых мощностей в облачных архитектурах "программное обеспечение как услуга".

Исследована проблема статистического мониторинга процессов с большими данными в компьютерных сетях, учитывающая свойства автокорреляции процессов образования и распада части локальных структур на ос-

нове контрольных диаграмм. Предложена оптимизационная модель и алгоритмы планирования пропускной способности облачных сервисов SaaS для нахождения компромисса между стоимостью ресурсов и штрафом за задержку выполнения. Осуществлена алгоритмизация определения минимальной верхней границы нулевого штрафа, обеспечивающий получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф.

Рис. 1. Дизайн исследования

Проанализированы требования к структуре программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга, обес-

печивающей принятие решений о допустимости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания. Результат анализа потребовал формализации данных задач, а также алгоритмизации их решения с учетом особенностей, отраженных на рис. 1. Сформулирована цель и задачи исследования.

Вторая глава посвящена оценке производительности сетевого мониторинга на основе разделяемых временных экспоненциальных моделей случайных графов.

Обозначим случайную сеть с n узлами через Y и ее наблюдение через y. Запишем граничную переменную между узлами i и j в виде Y_{ij} (i, j)$\in${1, 2, …, n}). В случае бинарных сетей, Y_{ij}=1 указывает на наличие ребра между i и j и Y_{ij}=0 представляет собой не-ребро. Обозначим количество ребер, 2-звездочек и треугольников в неориентированной сети Y через $S_1(Y)$, $S_2(Y)$ и T(Y). Эти численные параметры рассчитываются так:

$$S_1(Y) = \sum_{1\le i<j\le n} Y_{ij} \text{ - число ребер; } S_2(Y) = \sum_{1\le i\le n} \binom{Y_{i+}}{2} \text{ - число 2-звезд} \quad (2.1)$$

$$T(Y) = \sum_{1\le i<j<h\le n} Y_{ij}Y_{ih}Y_{jh} \text{ - число треугольников; } Y_{i+} \text{ - степень узла i}$$

Тогда модель ERGM экспоненциального случайного графа для случайной сети Y записывается так:

$$P(Y = y \mid \theta) = \frac{e^{\theta' h(y)}}{k(\theta, \Upsilon)} \quad (2.2)$$

где θ - вектор параметров, соответствующий статистическому вектору h(y); Υ обозначает пространство, содержащее все возможные наблюдения случайной сети Y; и $k(\theta, \Upsilon) = \sum_{y^* \in \Upsilon} e^{\theta' h(y^*)}$ - нормализующая константа.

Генерируя случайные наблюдения из ERGM по модели (2.2), можно получить не зависящую от времени сетевую последовательность. Для двух независимых сетей в последовательные моменты времени вероятности структурной статистики одинаковы, а расположение локальных структур полностью случайно. Переход от Y^t к Y^{t+1} происходит через образование и распад локальных структур.

Обозначим сетевой временной ряд через {Y^1, Y^2, …, Y^t, Y^{t+1},…}. Автокоррелированный сетевой процесс можно рассматривать как комбинацию двух промежуточных процессов, т.е. образования и распада части локальных структур, при этом остальная часть остается неизменной.

Благодаря Y^+ и Y^- процессы образования и распада могут быть непосредственно смоделированы как два ERGM:

$$P(Y^+ = y^+ \mid Y^t = y^t; \theta^+) = \frac{e^{\theta^+ h^+(y^+)}}{k(\theta^+, \Upsilon^+(y^t))}$$
$$P(Y^- = y^- \mid Y^t = y^t; \theta^-) = \frac{e^{(\theta^- h^-(y^-))}}{k(\theta^-, \Upsilon^-(y^t))} \quad (2.3)$$

$Y^+=Y^t\cup Y^{t+1}$ и $Y^-=Y^t\cap Y^{t+1}$ есть случайные промежуточные сети и y^+ и y^- есть их реализации; θ^+ и θ^- есть параметры образования и распада соответственно; $\Upsilon^+(y^t)=\{y': y^t\in y'\}$ обозначает пространство всех возможных сетей, содержащих y^t как подмножество, и $\Upsilon^-(y^t)=\{y': y'\in y^t\}$ обозначает пространство всех возможных сетей, которые являются подмножеством в y^t; $k(\theta^+,\Upsilon^+(t))$ и $k(\theta^-,\Upsilon^-(t))$ есть нормализующие константы.

Параметры создания и распада указывают на тенденцию соответствующих структур к новому созданию и распаду с течением времени. Следовательно, увеличение параметров создания при неизменных параметрах распада приводит к увеличению количества локальных структур, а уменьшение параметров формирования приводит к уменьшению количества локальных структур. Путем корректировки значений параметров моделируются сдвиги относительно уровней плотности, взаимности, изменчивости степени и свойств транзитивности сети.

Генерация сетей из STERGM определяется тремя составляющими: параметрами образования и распада θ^+ и θ^-, а также сетью y^t, на основе которой генерируется y^{t+1}. Вероятность приближения к совместному распределению вероятностей набора случайных величин равна

$$l(\theta)=\sum_{t=1}^{T}\sum_{i,j}\ln\left[\left(\pi_{ij}^t(\theta)\right)^{Y_{ij}^t}\left(1-\pi_{ij}^t(\theta)\right)^{1-Y_{ij}^t}\right] \qquad (2.4)$$

$$\hat{\theta}^+=\arg\max_{\theta^+} l(\theta^+),\ \hat{\theta}^-=\arg\max_{\theta^-} l(\theta^-).$$

Параметры образования и распада θ^+ и θ^- могут быть оценены на основе заданных выборок сети. Также временные ряды сети могут быть сгенерированы из модели (2.3) с заданными θ^+ и θ^-. Сеть в момент времени t+1 получается путем объединения двух промежуточных сетей в виде

$y^{t+1}=y^+ - (y^t - y^-)=y^-\cup(y^+ - y^t)$.

Хотя для мониторинга сетей с более сложной структурой было предложено много передовых методов, работа сосредоточена на контрольных диаграммах для самого основного свойства сети, т.е. плотности. Может быть четко выявлена закономерность автокорреляции, влияющая на показатели мониторинга. Контрольные диаграммы Шухарта, EWMA и CUSUM часто использовались в исследованиях по мониторингу сети.

При статистическом мониторинге процессов ожидается, что хорошо выполненная контрольная схема (1) вызовет ложную тревогу с очень малой вероятностью в состоянии "под контролем" и (2) своевременно обнаружит сдвиги в процессе. Обычно используемым критерием для оценки производительности является средняя продолжительность выполнения (ARL) контрольной диаграммы, которая сигнализирует о выбросе. ARL контрольной диаграммы в статусах "под контролем" и "вне контроля" обычно обозначаются ARL_0 и ARL_1. Ожидается, что значение ARL_0 контрольной диаграммы будет большим, в то время как значение ARL_1 будет небольшим. Сосредоточимся на изменении плотности в неориентированной сети и смоделируем се-

ти со сдвигами количества ребер. Уровень автокорреляции регулируется с помощью параметра деградации θ^-, а количество ребер поддерживается на целевом уровне с помощью параметра формирования θ^+. Запишем общее количество пар узлов как $C_n^2 = n(n-1)/2$. Параметр формирования ребер задается как

$$\theta^+ = -\log\left(\frac{1+e^{\theta^-}}{m/(C_n^2 - m)} - 1\right) \tag{2.5}$$

для достижения ожидаемого количества ребер m.

Чтобы исследовать влияние автокорреляций на контрольные диаграммы сети, используем платформу моделирования на основе STERGM для оценки характеристик контрольных диаграмм Шухарта, EWMA и CUSUM для подсчета ребер в сценариях низкого, среднего и высокого уровней автокорреляции, а также в независимом сценарии. Для контрольных диаграмм EWMA для исследования используются обычно используемые константы 0,05, 0,10 и 0,20 для λ. Для контрольных диаграмм CUSUM используем C_t^+ и C_t^- статистику для отслеживания положительных и отрицательных сдвигов соответственно. Предложенная система моделирования на основе STERGM была применена для изучения влияния автокорреляции на характеристики диаграмм Шухарта, EWMA, CUSUM и остаточного контроля для подсчета ребер. Результаты численных экспериментов показали, что автокорреляции оказывают неблагоприятное влияние на показатели контрольных диаграмм Шухарта, EWMA и CUSUM. Для контрольных диаграмм EWMA следует выбирать меньшее значение при наличии низкой и средней автокорреляции, в то время как в противном случае предпочтительнее большее значение. Как правило, контрольные диаграммы CUSUM лучше обнаруживают малые и средние сдвиги в сценариях с низкой и средней автокорреляцией. При наличии относительно высоких автокорреляций мы предлагаем использовать остаточные контрольные диаграммы, поскольку они лучше всего обнаруживают средние и большие сдвиги.

Таким образом, представлена модель статистического мониторинга процессов с большими данными в компьютерных сетях, отличающаяся учетом свойств автокорреляции процессов образования и распада части локальных структур на основе контрольных диаграмм и обеспечивающая моделирование сдвигов относительно уровней плотности, взаимности, изменчивости степени и свойств транзитивности сети.

Третья глава посвящена алгоритмизации мониторинга производительности коллективов встраиваемых объектов больших программных систем на основе временных рядов для анализа энергопотребления.

Предлагаемый метод анализа состоит из 3 этапов: сбор данных, создание объектно-ориентированной модели, поведенческая интерпретация этой модели. Исходные данные (выборки) преобразуются и агрегируются в иерархические объекты с учетом некоторых заданных свойств/показателей. В ре-

зультате получается высокоуровневая графовая модель, которая облегчает извлечение нетривиальных сведений о поведении системы, в частности о состояниях активности устройств и их последовательности, связанных с потреблением энергии. Это полезно для оценки системы, а также для ее коррекции или уточнения. Временной ряд - это последовательность из N выборок данных TS={$sample_1$, ..., $sample_N$}, которые записываются во временном порядке с фиксированными интервалами. Каждая выборка характеризуется своим значением и временной меткой. Измерительные устройства могут выдавать эти выборки в виде непрерывного потока данных или в виде пакетов с явной или неявной временной меткой (например, в зависимости от периода выборки и начальной временной метки для пакета). TS разбивается на M=N/n меньших частей, называемых сегментами (обозначаемыми как seg), содержащих заданное количество (n) последовательных выборок segi={$sample_{i\cdot n+1}$, ... $sample_{i\cdot n+n}$}. Каждый сегмент характеризуется набором параметров, рассчитанных по всем его выборкам, например: минимальный (min), максимальный (Max), средний (Av), среднеквадратичный (RMS), стандартное отклонение (SD), квартили и т.д. Последующие сегменты помечаются последующими значениями индекса. Некоторые из последующих сегментов могут быть очень похожи (с точки зрения их показателей). Поэтому целесообразно объединить их в группы. Группа (gk) - это последовательность последующих сегментов, удовлетворяющая следующему условию:

$$g_k = \{seg_i : \forall seg_i \in g_k \exists seg_{i+1} : |\, metric(seg_i) - metric(seg_{i+1}) \,| < \varepsilon_g \qquad (3.1)$$

где metric обозначает выбранные параметры сегмента (например, Av или SD), $i \in [1, \|g_k\|)$, а ε_g - предопределенная максимально допустимая разница показателей соседних сегментов в группе. Также могут быть определены другие показатели, определяющие более сложные условия, связанные с набором объектов. Сегменты, не удовлетворяющие условию (3.1), образуют отдельные группы сегментов. Описанный процесс группировки обеспечивает последовательность групп. Он основан на агрегировании последующих и сходных – с точки зрения заданного показателя – сегментов выборок TS. Эта последовательность строго связана с изменениями состояния системы. На данном этапе анализа TS может быть представлена в виде хронологически упорядоченных последовательностей групп. Для повышения точности модели границы между группами могут быть уточнены путем разделения соседних сегментов. Для каждой группы g_k может быть присвоено системное состояние s_x. Правильная идентификация групп, представляющих одно и то же состояние системы в пределах их последовательности, имеет решающее значение для анализа. Хотя все похожие группы в TS связаны с одним и тем же состоянием системы, данное состояние s_x (с меткой x) может быть определено как совокупность всех групп, удовлетворяющих следующему условию:

$$s_x = \{g_i : \forall g_i \in s_x \; \exists g_j \in s_x, i \neq j \wedge |\, metric(g_i) - metric(g_j) \,| < \varepsilon_s \quad (3.22)$$

где ε_s - заданная максимально допустимая разница показателей, относящихся к рассматриваемым группам. Группы, не удовлетворяющие условию (2), от-

носятся к состояниям одной группы. Также возможны более сложные условия, включающие несколько показателей и различные свойства. Для данного временного ряда вводится ориентированный граф состояний SG={s_{g0}, S, E}, где s_{g0} - это состояние начальной группы TS (относящееся к исходному состоянию системы), S и E - множества вершин (т.е. все состояния) и ребер между вершинами (узлы), соответственно. Набор ребер определяется следующим образом:

$$E = \{(u,v) \in S, u \neq v, \exists g_i \in u \wedge g_j \in v : \exists seg_k \in g_i \wedge seg_l \in g_j, k+1=l\} \quad (3.3)$$

где i, j $\in$ [0, |G|), G - множество всех обнаруженных групп, а k, l$\in$ [0, M). Это условие гарантирует, что ребро соединяет соседние (по времени) группы в TS, которые принадлежат, соответственно, двум разным состояниям u и v. Сегмент seg_k является последним в группе, принадлежащей состоянию u, в то время как сегмент seg_l является первым в группе, принадлежащей состоянию v. Для каждого состояния определены наборы выходных и входных ребер s_x:

$E_{out(x)}$={(s_x,v): (s_x,v)$\in$E} (3.4)

$E_{in(x)}$={(u, s_x): (u, s_x)$\in$E} (3.5)

В зависимости от целей анализа различают конкретные состояния: последовательные состояния, удовлетворяющие условию $|E_{out(x)}|=|E_{in(x)}|=1$, и базовое состояние ($s_B$) с предопределенными свойствами, например, с множеством соответствующих ребер, т.е.:

$$\exists s_b, \forall s_i \in S, \left|E_{out(B)}\right| + \left|E_{in(B)}\right| > \left|E_{out(i)}\right| + \left|E_{in(i)}\right| \quad (3.6)$$

С точки зрения встроенных систем, базовое состояние соответствует состоянию, из которого выполняются все системные действия и в которое система переходит после этих действий (т.е. основному циклу). После определения таких состояний анализ может быть сосредоточен на действиях, выполняемых системой в ответ на различные события (синхронные и асинхронные – таймеры, прерывания и т.д.). Цикл (C) представляет собой последовательность групп, для которых последующие элементы соединены ребром в системном графе (SG), и первое состояние в этой последовательности соединено ребром, принадлежащим $E_{out(B)}$, а последнее - ребром, принадлежащим $E_{in(B)}$, где s_B - это базовое состояние. Цикл C={g_i, ..., g_p}, где {g_i, ..., g_p}$\in$G и i...p - последовательные номера групп (идентификаторы), удовлетворяет следующим условиям:

1) $(g_i \in s_i) \wedge (s_B, s_i) \in E_{out(B)}; (g_p \in s_p) \wedge (s_p, s_B) \in E_{in(B)}$

2) пусть s^{g_x} обозначает состояние, к которому принадлежит группа g_x: $\forall x \in [i,p) : g_x, g_{x+1} \in C, (s^{g_x}, s^{g_{x+1}}) \in E, vs^{g_x} = s^{g_{x+1}}$

3) $\forall x \in [i,p) : s^{g_x} \neq s_B$

Разработано 12 алгоритмов управления мониторингом производительности коллективов встраиваемых объектов больших программных систем на основе временных рядов для анализа энергопотребления. Наиболее значимые приведены ниже.

Алгоритм 3.1. Определение состояний

Input: *Groups* – список групп сегментов
Output: список выявленных состояний

```
1: function detect_states(Groups)
2: States=new list
3: foreach group in Groups do
4:     state=add_to_similar_state(States, group)
5:     if (state is None) then
6:         state=create_new_state(group)
7:         States.add(state)
8:     end if
9: end foreach
10: return States
11: end function
```

Алгоритм 3.2. Определение ребер графа состояния системы

Input: *Groups* – список последующих групп с присвоенными метками состояний
Output: *Edges* – набор границ между состояниями

```
1: function detect_edges(Groups)
2: Edges=new set
3: foreach group in Groups do
4:     If (group.next is not None) then
5:         if (group.StateLabel != group.next.StateLabel) then
6:             Edges.add(new Edge(group.StateLabel, group.next.StateLabel))
7:         end if
8:     end if
9: end foreach
10: return Edges
11: end function
```

На практике производный граф состояний может быть довольно сложным, в частности, содержать длинные последовательности состояний с одним входным и выходным ребрами. Но во многих случаях нецелесообразно сосредотачиваться на них – упрощение графика состояний путем объединения последовательности загрузки в одно состояние может облегчить дальнейший анализ. Следовательно, может быть создан сокращенный граф, который объединяет такие последовательности состояний в уникальное обобщенное состояние (в зависимости от целей анализа). Этот процесс реализуется с помощью процедуры merge_simple_sequences() (алгоритм 3.3).

Выбор параметров для алгоритмов выполняется тремя способами:

1) на основе предыдущего опыта анализа (например, взятие параметров из ранее проанализированной версии проекта для проверки свойств следующей версии),

2) на основе практического опыта эксперта, который знает ожидаемые классы циклов (их периодичность, уровни мощности и т.д.),

3) итеративный поиск в экспериментах с репрезентативным фрагментом изучаемой TS или полным фрагментом.

Алгоритм 3.3. Объединение простых последовательностей

```
Input: graph – граф
Output: graph – граф с объединенными простыми
последовательностями (измененные узлы и состояния)
1: function merge_simple_sequences(graph)
2:     graph.Nodes.clear_visited_flags()
3:     sequences=new set
4:     current_sequence=new list
5:     identify_simple_sequences(graph.Nodes.first, sequences,
current_sequence)
6:     merge_states_and_nodes(sequences, graph)
7: end function
```

Алгоритм 3.4. Идентификация простых последовательностей

```
Input: node – объект узла графа, sequences – набор
       последовательностей, current_sequence – обрабатываемая в
       данный момент последовательность
Output: sequences – обновленный набор последовательностей
1: function identify_simple_sequences(node, sequences,
     current_sequence)
2: node.visited=True
3: if (node.In_edges.count <= 1 and node.Out_edges.count <= 1) then
4:     current_sequence.add(node)
5:     if (node.Out_edges.count == 0) then
6:         sequences.add(current_sequence)
7:         current_sequence=new list
8:     end if
9:     else
10:        if (current_sequence.Length > 1) then
11:            sequences.add(current_sequence)
12:            current_sequence=new list
13:        end if
14:    end if
15:    foreach neighbour in node.Neighbours do
16:        if (neighbour.visited is False) then
17:        identify_simple_sequences(neighbour, sequences,
           current_sequence)
18:        end if
19:    end foreach
20: end function
```

Для TS мониторинга программных систем учета энергопотребления сгенерированный граф состояний содержал 99 состояний. Примерно для 50 из этих состояний количество входных и выходных ребер было равно 1, следовательно, при использовании алгоритма 3.4 первичный граф был сокращен до 49 состояний и показан на рис. 2.

Таким образом, предложены алгоритмы мониторинга производительности коллективов встраиваемых объектов больших программных систем, отличающиеся предварительной кластеризацией и идентификацией поведенческих последовательностей, удовлетворяющих условиям сходства, обеспечивающие возможность иерархического анализа данных и улучшающие интерпретируемость результатов.

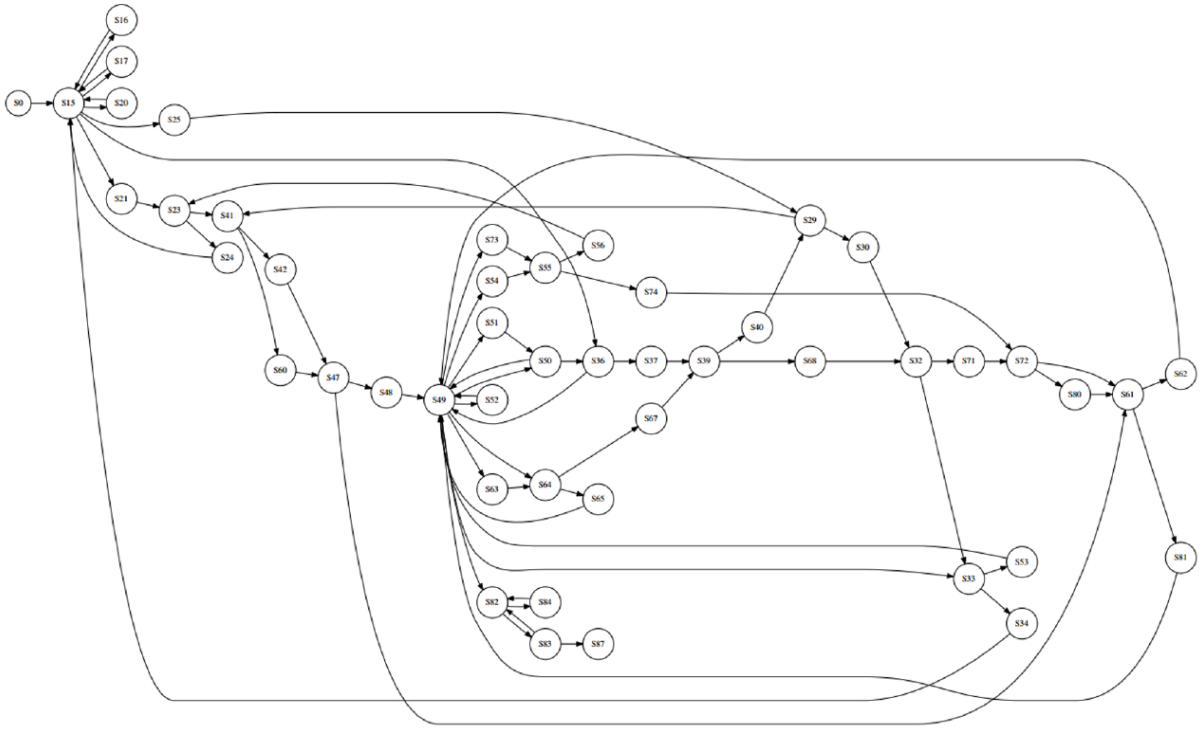

Рис. 2. Граф состояний анализируемого временного ряда

В главе 4 представлены новые результаты в области управления эластичностью для планирования мощностей в облачных средах типа "Программное обеспечение как услуга", обслуживающих системы мониторинга.

Поставщики облачных услуг используют планирование производительности для поддержания производительности вычислительных ресурсов (экземпляров), необходимой для удовлетворения динамического характера спроса (запросов). Однако существует компромисс между развертыванием слишком большого количества дорогостоящих экземпляров и развертыванием слишком малого количества экземпляров и выплатой штрафных санкций за невозможность своевременной обработки запросов. Определение оптимального количества экземпляров, необходимого для данного горизонта планирования, является сложной задачей из-за комбинаторного характера задачи оптимизации.

Определим горизонт планирования как период времени, в течение которого начисляются штрафы за задержку запроса, поэтому он выбирается в качестве основного ориентира модели. Чтобы определить штрафные санкции за задержку для конкретного клиента, облачные провайдеры сравнивают среднее время ответа для конкретного типа запроса с пороговым значением штрафной задержки для конкретного типа запроса на каждом горизонте планирования. В различных приложениях запросы могут поступать в любое время и в любой комбинации. Определим эту непрерывную общую задачу для заданного пути выборки как задачу DQ_g.

Общая задача DQ_g сформулирована с целым числом γ_t^j запросов типа j, поступающих в течение временного интервала t. Учитывая DQ_g, преобразуем ее в задачу DQ_1, перемещая все запросы, поступающие в течение каждого

временного интервала t, в начало временного интервала t. Пусть оптимальным решением задачи DQ_g будет Π_g при значениях решения (X_i^g, Y_{it}^{gj}), а оптимальным решением задачи DQ_1 будет Π при значениях решения (X_i, Y_{it}^j).

В работе доказано

Предположение 4.1. Решение задачи DQ_1 является нижней границей общей задачи DQ_g: $\Pi_g \geq \Pi$.

Используем смешанную постановку для модели оптимизации задачи DQ_1, поскольку количество экземпляров может быть только целым, но штрафные условия непрерывны. Основной переменной для принятия решения является количество развернутых экземпляров $\sum_{i=1}^{m} X_i$, где $X_i=1$, если экземпляр i развернут в течение периода планирования; в противном случае $X_i=0$. Задача DQ_1 имеет следующую формулировку (обозначения приведены в табл. 4.2 диссертации):

$$\text{Min } \Pi\left(X_i, Y_{it}^j\right) = \sum_{j=1}^{k} Q_j + \alpha \sum_{i=1}^{m} X_i \tag{4.0}$$

с ограничениями:

$$I_t^j = I_{t-1}^j + \gamma_t^j - \sum_{i=1}^{m} Y_{it}^j,\ j = 1,2,...,k;\ t = 1,2,...,n \tag{4.1}$$

$$Y_{it}^j \leq u_j X_i,\ j = 1,2,...,m;\ j = 1,2,...,k;\ t = 1,2,...,n \tag{4.2}$$

$$L_{it}^j \geq \sum_{h=0}^{\lambda_j - 1} Y_{i,t-h}^j,\ j = 1,2,...,k;\ i = 1,2,...m;\ t = 1,2,...,n \tag{4.3}$$

$$\sum_{j=1}^{k} c_r^j L_{it}^j \leq C_r X_i,\ r = 1,2,3;\ i = 1,2,...m;\ t = 1,2,...,n \tag{4.4}$$

$$P_j \geq \frac{\sum_{t=0}^{n} \tau I_t^j}{u_j},\ j = 1,2,...,k \tag{4.5}$$

$$P_j + \tau\lambda_j \leq \tau\mu_j + V_j M,\ j = 1,2,...,k \tag{4.6}$$

$$Q_j \geq b_j u_j (P_j + \tau\lambda_j - \tau\mu_j) - (1 - V_j)M,\ j = 1,2,...,k \tag{4.7}$$

$$Y_{it}^j, I_t^j, L_{it}^j > 0,\ i = 1,2,...,m;\ j = 1,2,...,k;\ t = 1,2,...n \tag{4.8}$$

$$P_j, Q_j \geq 0,\ j = 1,2,...,k \tag{4.9}$$

$$X_i, V_j \in \{0,1\},\ i = 1,2,...,m;\ j = 1,2,...,k \tag{4.10}$$

Задача DQ_1 отличается от традиционных задач с упаковкой в ячейки, поскольку запросы поступают динамически в разные периоды и должны быть упакованы в экземпляры (ячейки) определенной емкости. Запрос покидает ячейку после выполнения, что создает пространство в ячейке для других запросов, ожидающих упаковки, а также целевые функции этих двух задач различны, и за превышение ожидания запроса в DQ_1 взимается штрафная плата за задержку. В работе доказаны следующие предположения.

Предположение 4.2. Оптимальное значение целевой функции Π^r задачи DQ_1^r не увеличивается с увеличением r при $r \le m$. Отсюда $\Pi^1 \ge \Pi^2 \ge \ldots \Pi^m \ge \Pi^{m+1}$, r=1, 2, …, m+1.

Предположение 4.3. В оптимальных решениях задачи $\overline{D}\overline{Q}_1^r$, r=1, 2, ..., m+1, существует индекс m^* такой, что $\overline{S}_q^1 > \overline{S}_q^2 > \ldots > \overline{S}_q^{m^*}$ и $\overline{S}_q^{m^*} = \overline{S}_q^{m^*+1} = \ldots = \overline{S}_q^m = \overline{S}_q^{m+1}$, где $1 \le m^* \le m$.

Предположение 4.4. $S_q^r \ge \overline{S}_q^r$, r=1, 2, …, m+1.

Предположение 4.5. $\Pi^{m_0-1} > \Pi^{m_0} = \Pi^{m_0+1} = \Pi^m = \Pi^{m+1}$ и $m_0 \le m^* \le m$.

На их основе доказано с использованием задачи о 3D-разбиении

Утверждение 4.1. Задача DQ_1 является сильно NP-сложной.

Поскольку оптимальное решение модели фиксированной емкости с r экземплярами не приводит к увеличению числа фиксированных экземпляров (предположение 4.2), для достижения решения обычно не требуется рассматривать все возможности фиксированных экземпляров.

Чтобы упростить структурированный стратегический подход, определим две отдельные задачи $(DQ_1^r, \overline{D}\overline{Q}_1^r)$, аналогичные задаче DQ_1, с дополнительными ограничениями на общее количество развернутых экземпляров $\sum_{i=1}^{m} X_i$. В $\overline{D}\overline{Q}_1^r$ число экземпляров фиксировано на уровне r, тогда как в DQ_1^r число экземпляров меньше или равно r. Для простоты обозначим оптимальное значение целевой функции $\Pi^r(X_i, Y_{it}^j)$ задачи DQ_1^r как Π_r, а соответствующее значение $\sum_{j=1}^{k} Q_j$ (соответственно, $\sum_{i=1}^{r} X_i$) - как S_q^r (соответственно, Z_x^r). Заметим, что $Z_x^r \le r$, $S_q^m = 0$ и $S_q^{m+1} = 0$. Кроме того, мы обозначаем оптимальное значение целевой функции $\sum_{j=1}^{k} Q_j$ задачи $\overline{D}\overline{Q}_1^r$ как $\overline{S}_q^r$. Поскольку $S_q^m = 0$ и $S_q^{m+1} = 0$, имеем $\overline{S}_q^m = 0$ и $\overline{S}_q^{m+1} = 0$. В структурированной стратегии решаем задачу $\overline{D}\overline{Q}_1^r$ с фиксированным числом экземпляров r, начиная со среднего значения, r=(1+m)/2. Затем итеративно увеличиваем или уменьшаем значение r, чтобы достичь улучшенного верхнего граничного значения, m*, как определено в предположении 4.3), а затем находим наименьшее количество экземпляров, которые приводят к минимальной общей стоимости, m_0, как определено в предположении 4.5.

Конкретные шаги этой структурированной стратегии описаны в алгоритме 4.1.

Алгоритм 4.1. Структурированная стратегия решения задачи DQ_1

1 **procedure Structured-Strategy** $(\overline{DQ}_1^r, \overline{DQ}_1^{r+1})$

2 $r = \left\lfloor \frac{1+m}{2} \right\rfloor$

3 Решаем задачи $(\overline{DQ}_1^r, \overline{DQ}_1^{r+1})$, находя решения $\overline{S}_q^r$, $\overline{S}_q^{r+1}$ соответственно

4 **while** $\overline{S}_q^r > \overline{S}_q^{r+1}$ **do**

5 r=r+1

6 Решаем задачу $\overline{DQ}_1^{r+1}$ и находим решение $\overline{S}_q^{r+1}$

7 **end while**

8 **while** $\overline{S}_q^r = \overline{S}_q^{r+1}$ **do**

9 r=r-1

10 Решаем задачу $\overline{DQ}_1^r$ и находим решение $\overline{S}_q^r$

11 **end while**

12 $m^* = r$

13 $m^0 = m^*$

14 Решаем задачи $DQ_1^{m_0-1}$, $DQ_1^{m_0}$ и находим решения Π^{m_0-1}, Π^{m_0} соответственно

15 **while** $\Pi^{m_0-1} = \Pi^{m_0}$ do

16: $m_0 = m_0 - 1$

17: Решаем задачу $DQ_1^{m_0-1}$ и находим решение Π^{m_0-1}

18: **end while**

19: **return** m_0, Π^{m_0}

20: **end procedure**

Таким образом, разработаны оптимизационная модель и алгоритмы планирования пропускной способности облачных сервисов SaaS, отличающиеся форсированным решением NP-сложной задачи минимизации затрат на задержку выполнения запросов и на ресурсы развернутых экземпляров и обеспечивающие нахождение компромисса между стоимостью ресурсов и штрафом за задержку выполнения во встраиваемых облачных приложениях "программное обеспечение как услуга" с точки зрения поставщика облачных услуг.

Также разрабатываются алгоритмы решения проблемы неопределенности спроса в задачах мониторинга.

Для решения задачи Алгоритм 4.3 с нулевым штрафом вызывает Алгоритм 4.2 назначения запросов в очереди.

Данные для сравнительно анализа получены из облачного веб-приложения для мониторинга объектов интернет-вещей, которое обслуживает тысячи пользователей клиента. Оптимизационная модель, а также алгоритмы реализованы C++. Для получения результатов оптимизации использован пакет CPLEX 12.7.1. Все реализации алгоритмов запускались на серверах с 24-ядерным процессором Intel(R) Xeon(R) CPU E5-2683 v4 с тактовой частотой 2,1 ГГц, 64 МБ кэш-памяти и 256 ГБ оперативной памяти, работающих под управлением операционной системы Ubuntu 16.04.3. Пример результатов моделирования представлен на рис. 3.

По результатам имитационного моделирования среднее снижение затрат при использовании алгоритма по сравнению с существующим составляет 4,73% и позволяет использовать меньше задействованных ресурсов.

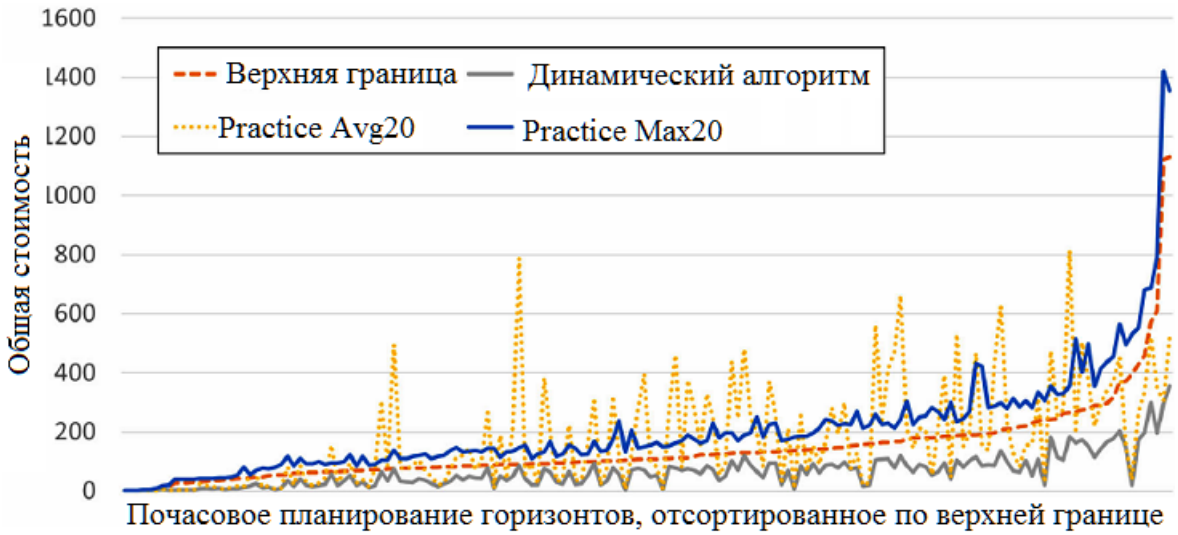

Рис. 3. Сравнение итоговых затрат целевой функции по верхней границе и OFD с текущими практическими подходами к управлению потенциалом горизонтальной эластичности

Алгоритм 4.2. Назначение запросов из очереди для начала выполнения

procedure Assign-Queries-In-Queue(H, n, k, γ_t^j)
  for t=1 **to** n **do**
    Добавляем все вновь поступающие запросы в очередь ожидания.
    Сортируем очередь запросов, сначала выполняя запросы с наименьшим временем обработки.
    for j=1 **to** k **do**
      Если запрос типа j ожидает, пробуем поместить его в один из его экземпляров. Если в экземпляре найдено место во всех трех измерениях емкости, добавим требования к емкости этого запроса для этого экземпляра на некоторое количество временных интервалов в будущем, равное времени его выполнения.
    end for
    Запишем все запросы, которые не могут быть назначены экземпляру, в очередь ожидания.
  end for
  Рассчитаем потенциальные штрафные санкции Q_j, используя наборы ограничений (4.5)-(4.7).
  return Q_j
end procedure

Таким образом, представлен динамический алгоритм определения минимальной верхней границы нулевого штрафа, отличающийся использованием ожидающей очереди неисполненных запросов и обеспечивающий получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф.

Далее разработана структура программного прототипа оптимизации облачных сервисов SaaS для решения задач мониторинга (рис. 4). Соответствующая архитектура программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга отличается интеграцией системы мониторинга и сервисов SaaS и обеспечивает принятие решений о допусти-

мости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания**.** Элементы программной реализации прошли государственную регистрацию в Роспатенте.

Алгоритм 4.3. Определение минимальной верхней границы нулевого штрафа

```
1 procedure Zero-Penalty (m, n, k, γ_t^j)
2     H=m
3     repeat
4         Call Assign-Queries-In-Queue
5         H=⌊H*(1−ZeroPenJump)⌋,
          где ZeroPenJump - процентное уменьшение
6     until Penalty > 0
7     repeat
8         H=H+1
9:        Call Assign-Queries-In-Queue в Алгоритме 4.2
10    until Penalty=0
11    m=H,
      минимальное количество случаев, необходимое для достижения
      нулевых штрафных санкций.
12    return m
13 end procedure
```

Таким образом, создана структура программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга, отличающаяся интеграцией системы мониторинга и сервисов SaaS и обеспечивающая принятие решений о допустимости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания.

Заключение

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Разработана модель статистического мониторинга процессов с большими данными в компьютерных сетях, учитывающая свойства автокорреляции процессов образования и распада части локальных структур на основе контрольных диаграмм.

2. Представлены модифицированные алгоритмы мониторинга производительности коллективов встраиваемых объектов больших программных систем, обеспечивающие возможности иерархического анализа данных и улучшающая интерпретируемость результатов.

3. Создана оптимизационная модель и алгоритмы планирования пропускной способности облачных сервисов SaaS, позволяющая найти компромисс между стоимостью ресурсов и штрафом за задержку выполнения во встраиваемых облачных приложениях "программное обеспечение как услуга" с точки зрения поставщика облачных услуг.

4. Разработан динамический алгоритм определения минимальной верхней границы нулевого штрафа, обеспечивающий получение верхней границы объема вычислительных ресурсов, еще дающей нулевой штраф. Среднее снижение затрат при использовании алгоритма по сравнению с существующим составляет 4,73% и позволяет использовать меньше задействованных

ресурсов.

5. Разработана структура программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга, обеспечивающая принятие решений о допустимости конфигурации сервиса с точки зрения штрафов, и также оценку ожидаемого качества обслуживания.

6. Компоненты программного обеспечения зарегистрированы в Роспатенте.

Рис. 4. Структура и функции прототипа программного обеспечения оптимизации облачных сервисов SaaS для решения задач мониторинга

Рекомендации и перспективы дальнейшей разработки темы

1. Результаты исследования рекомендуются к применению в задачах

управления процессами мониторинга распределенного программного обеспечения.

2. Дальнейшая разработка темы будет направлена на практическую реализацию теоретических и алгоритмических результатов, интеграцию в проекты наиболее распространенных распределенных систем.

Основные результаты диссертации опубликованы в следующих работах:

Публикации в изданиях списка ВАК

1. Сидоренко Е.В., Рындин Н.А., Корнеев А.М. Рационализация выбора оптимальной виртуальной машины для реализации специальных запросов в облачной среде// Системы управления и информационные технологии, №4(94), 2023. С. 41-45

2. Варламов О.О., Чжан С., Балдин А.В., Мышенков К.С., Сидоренко Е.В. Разработка миварной экспертной системы для планирования ресурсов цеха и анализа отклонений. Моделирование, оптимизация и информационные технологии. 2024; 12(3). https://moitvivt.ru/ru/journal/pdf?id=1641. DOI: 10.26102/2310-6018/2024.46.3.017.

3. Сидоренко Е.В. Аппаратно-программное обеспечение средств управления большими данными наблюдаемых энергетических систем// Системы управления и информационные технологии, №4(98), 2024. С. 92-98.

4. Амоа Куадио-кан Армел Жеафруа, Сидоренко Е.В., Рындин Н.А. Мониторинг состояния коммуникационных сетей на основе облачных вычислений в режиме реального времени// Моделирование, оптимизация и информационные технологии. 2025;13(1). URL: https://moitvivt.ru/ru/journal/pdf?id=1809. DOI: 10.26102/2310-6018/2025.48.1.014.

Публикации в изданиях, индексируемых в Scopus и WoS

5. Sidorenko E.V. et al. Elements realization of software algorithmic system optimization of technological processes based on adaptive methods// IOP Conference series: Materials science and engineering, 2020, 791(1):012034 **(Scopus)**.

6. Sidorenko E.V. et al. Designing the architecture of a distributed system for information monitoring of IoT and IIoT infrastructures traffic// International Journal on Information Technologies and Security, vol. 16, no. 1, 2024, pp. 49-56. https://doi.org/10.59035/BTBI7690 **(WoS)**.

Свидетельство о государственной регистрации программы для ЭВМ

7. Программа интерактивного управления очередью системы мониторинга/ Е.В. Сидоренко, М.В. Кочегаров, В.А.К. Камиль, К.А.Ж. Амоа, О.А. Ющенко. Свидетельство о регистрации программы для ЭВМ № 2025615513 от 12.02.2025. М.: ФИПС, 2025.

Статьи и материалы конференций

8. Сидоренко Е.В. и др. Реализация элементов программно-алгоритмического комплекса оптимизации технологических процессов на базе адаптивных методов// Энергетические системы. – 2019.– № 1. – С. 169-175.

9. Сидоренко Е.В. Проблемы мониторинга сетевой инфраструктуры на основе типовых решений// Информационные технологии моделирования и

управления, №4(134), 2023. – С. 311-320.

10. Sidorenko E.V. An overview of methods for evaluating network monitoring performance based on shared time exponential random graph models// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2024'AS): Proc. of the XXIX-th Int. Open Science Conf. - Yelm, WA, USA: Science Book Publishing House, 2024. Pp. 216-222.

11. Сидоренко Е.В. Моделирование экспоненциального случайного графа для сетей, не зависящих от времени как основа метода оценки производительности// Информационные технологии моделирования и управления, №1(135), 2024. – С. 52-60.

12. Сидоренко Е.В., Рындин Н.А. Результаты влияния автокорреляций на контрольные диаграммы мониторируемой сети // Наука и технологии: перспективы развития и применения: сб. статей VII Междунар. НПК. - Петрозаводск: МЦНП «НОВАЯ НАУКА», 2024. - С. 23-30.

13. Sidorenko E.V. et al. Algorithmization of time series processing as the results of the work of teams of embedded objects of large software systems of the monitoring subsystem// Modern informatization problems in simulation and social technologies (MIP-2025'SCT): Proc. of the XXX-th Int. Open Science Conf. - Yelm, WA, USA: Science Book Publishing House, 2025. – pp. 96-110.

Подписано в печать 25.04.25
Формат 60x84/16. Бумага для множительных аппаратов.
Усл. печ. л. 1,0. Тираж 80 экз. Заказ №196.
ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14