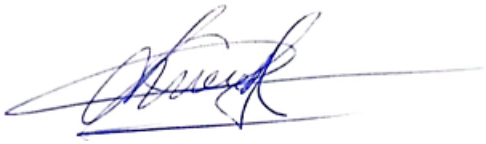
На правах рукописи



Амоа Куадио-кан Армел Жеафруа

УПРАВЛЕНИЕ ПРОЦЕССАМИ РАЗРАБОТКИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ 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: **Алексеев Владимир Витальевич**, доктор технических наук, профессор, «Тамбовский государственный технический университет», профессор кафедры «Информационные системы и защита информации»

Семенова Елена Георгиевна, доктор технических наук, профессор, ФГБОУ ВО «Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова», г. Санкт-Петербург, заведующая кафедрой О7 «Информационные системы и программная инженерия»

Ведущая организация: **ФГБОУ ВО "Сибирский государственный университет науки и технологий имени академика Михаила Фёдоровича Решетнёва», г. Красноярск**

Защита состоится «27» июня 2025 года в 14^{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. Инфокоммуникационные системы являются базовыми в большинстве значимых аспектов функционирования современного общества. В этой связи особенно важной является задача эффективно управления моделированием и разработкой программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных архитектурах. В рамках жизненного цикла такого программного обеспечения одна из подзадач связана с необходимостью ситуационного моделирования и анализа организации-разработчика специального программного обеспечения с открытым программным кодом. В конечном счете табличный учет каждого варианта открытости и учет специфики предметной области создаваемого продукта на основе горизонтальных или диагональных связей позволяет разрабатывать и распространять программное обеспечение максимально эффективно. Большой вклад в развитие методов моделирования и разработки программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных архитектурах внесли Ковалев И.В., Кравец О.Я., Лукьянчиков О.И., Никульчев Е.В., Ciardo G., Gribaudo M.I., Jakobik A. Разумной идеей кажется создание ситуационной модели организации-разработчика программного обеспечения с открытым программным кодом, обеспечивающей определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

Исследование облачных архитектур вследствие практической безграничности объекта исследования невозможно без моделирования, особенно с учетом высокой стохастичности происходящих процессов. Методы аналитического исследования, к сожалению, еще не развиты в достаточной степени, поэтому на первый план выступают имитационные исследования. Как следствие, особый интерес представляют технологии стохастического моделирования облачной архитектуры, позволяющую автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с поведением облачной архитектуры.

Важным этапом управления инфокоммуникационными системами в облачных средах является идентификация состояния инфокоммуникационной системы на основе облачных вычислений, обеспечивающая расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки. Разумеется, при этом важен анализ производительности и прогнозирование доступности многоуровневой облачной среды, обеспечивающую учет времени жизни основных компонентов и оценку доступности облачной среды.

Таким образом, актуальность темы диссертационного исследования продиктована необходимостью развития моделей анализа и алгоритмов разработки специального программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных средах на основе ситуационной обработки каждого варианта открытости и учетом специфики предметной области продукта на основе горизонтальных или диагональных связей.

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

Целью работы является создание моделей анализа и алгоритмов разработки специального программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных средах.

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

1. Проанализировать проблематику сквозного моделирования и алгоритмизации процессов разработки специального программного обеспечения с открытым кодом в облачных средах.
2. Разработать ситуационную модель анализа организации-разработчика программного обеспечения с открытым программным кодом, обеспечивающую определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.
3. Предложить модификацию технологии стохастического моделирования облачной архитектуры, позволяющую автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с поведением облачной архитектуры
4. Разработать алгоритм идентификации состояния инфокоммуникационной системы на основе облачных вычислений, обеспечивающий расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки
5. Разработать модель анализа производительности и прогнозирования доступности многоуровневой облачной среды, обеспечивающую учет времени жизни основных компонентов и оценку доступности облачной среды.
6. Разработать структуру специального программного обеспечения распознавания блокировок и оптимизации доступности облачных сервисов, обеспечивающую реконфигурацию инфокоммуникационной системы в зависимости от параметров инфраструктуры и качества обслуживания.

Объект исследования: процессы построения моделей анализа и алгоритмов разработки специального программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных средах.

Предмет исследования: средства математического и программного управления процессами анализа облачной архитектуры и производительности инфокоммуникационных систем на основе алгоритмов генерации имитационных моделей и архитектуры системы распознавании состояний блокировок.

Методы исследования. При решении поставленных в диссертации задач использовались методы теории графов, теории вероятностей, теории принятия решений, теории моделирования, 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»; п.10 «Оценка качества, стандартизация и сопровождение программных систем».

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

1. Ситуационная модель анализа организации-разработчика программного обеспечения с открытым программным кодом, отличающаяся табличным учетом каждого варианта открытости и учетом специфики предметной области продукта на основе горизонтальных или диагональных связей и обеспечивающая определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

2. Модификация технологии стохастического моделирования облачной архитектуры, отличающаяся использованием мультиформального подхода для определения классов рабочей нагрузки, конфигурации облака и взаимосвязей между рабочей нагрузкой и облаком в сценариях мультиоблачных/гибридных облачных/пограничных вычислений, позволяющая автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с поведением облачной архитектуры.

3. Алгоритм идентификации состояния инфокоммуникационной системы на основе облачных вычислений, отличающийся событийным мониторингом ситуаций блокировки системы и обеспечивающий расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки

4. Модель анализа производительности и прогнозирования доступности многоуровневой облачной среды, отличающаяся периодическим расчетом времени жизни облачной системы и обеспечивающая учет времени жизни основных компонентов и оценку доступности облачной среды.

5. Структура комплекса программного обеспечения процесса распознавания блокировок и оптимизации доступности облачных сервисов, отличающаяся реализацией механизмов интеграции системы мониторинга и облачных сервисов и обеспечивающая реконфигурацию инфокоммуникационной системы в зависимости от параметров инфраструктуры и качества обслуживания.

Теоретическая и практическая значимость исследования заключается в развитии специальных средств разработки специального программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных средах на основе ситуационной обработки каждого варианта открытости и учетом специфики предметной области продукта на основе горизонтальных или диагональных связей.

Теоретические результаты работы могут быть использованы в проектных и научно-исследовательских организациях, занимающихся разработкой программного обеспечения инфокоммуникационных систем с открытым исходным кодом в облачных средах.

Положения, выносимые на защиту

1. Ситуационная модель анализа организации-разработчика программного обеспечения с открытым программным кодом обеспечивает определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

2. Модификация технологии стохастического моделирования облачной архитектуры позволяет автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с поведением облачной архитектуры

3. Алгоритм идентификации состояния инфокоммуникационной системы на основе облачных вычислений обеспечивает расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки.

4. Модель анализа производительности и прогнозирования доступности многоуровневой облачной среды обеспечивает учет времени жизни основных компонентов и оценку доступности облачной среды.

5. Структура программного обеспечения распознавания блокировок и оптимизации доступности облачных сервисов позволяет осуществить реконфигурацию инфокоммуникационной системы в зависимости от параметров инфраструктуры и качества обслуживания.

Результаты внедрения. Основные результаты работы внедрены в ООО «Центр информационных технологий» (г. Воронеж) при проектировании доступной облачной системы управления инфокоммуникационными сервисами, 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: VII Международной НПК «Наука и технологии: перспективы развития и применения» (Петрозаводск, 2024); VI Всероссийской НПК «Информационные технологии в экономике и управлении» (Махачкала, 2024); XXX International Open Science Conference «Modern informatization problems in the technological and telecommunication systems analysis and synthesis» (Yelm, WA,, USA, 2025), а также на научных семинарах кафедр САПРИС и искусственного интеллекта и цифровых технологий ВГТУ (2019-2025 гг.).

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

Публикации. По результатам диссертационного исследования опубликовано 12 научных работ (5 – без соавторов), в том числе 5 – в изданиях, рекомендованных ВАК РФ (из них 1 – в издании Wos и одно свидетельство о регистрации программы для ЭВМ). В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [2, 12] - технология стохастического моделирования облачной архитектуры на основе мультиформального подхода; [3] - алгоритм идентификации состояния инфокоммуникационной системы на основе облачных вычислений; [8] - ситуационная модель организации-разработчика программного обеспечения с открытым программным кодом; [4] - структура программного обеспечения распознавания блокировок и оптимизации доступности облачных сервисов; [1, 5] - информационное и программное обеспечение для экспериментальной оценки качества разработанных методов и алгоритмов.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 169 наименований. Работа изложена на 167 страницах.

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 особенности математического и программного управления процессами исследования облачных архитектур и производительности инфокоммуникационных систем на основе алгоритмов генерации имитационных моделей и архитектуры системы распознавании состояний блокировок. Отмечено, что повысить эффективность управления можно путем разработки ситуационной модели организации-разработчика программного обеспечения с открытым программным кодом, обеспечивающая определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

Исследована проблема автоматической генерации стохастических имитационных моделей с высоким уровнем согласованности с поведением облачной архитектуры. Определен мультиформальный метод, позволяющий интегрировать ситуационный подход к высокоуровневому моделированию с известным симулятором облачных архитектур для получения более реалистичных результатов в процессе анализа производительности. Этот метод призван заполнить пробел между предварительными (аналитическими) моделями анализа производительности и подробными имитационными моделями на основе поэтапного моделирования.

Далее необходимо провести исследование и разработку алгоритма идентификации состояния инфокоммуникационной системы на основе облачных вычислений, обеспечивающего расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки.

Затем нужно решить задачу построения модели анализа производительности и прогнозирования доступности многоуровневой облачной среды, обеспечивающей учет времени жизни основных компонентов и оценку доступности облачной среды.

Наконец, решение задачи проектирования структуры программного обеспечения распознавания блокировок и оптимизации доступности облачных сервисов, обеспечивающей реконфигурацию инфокоммуникационной системы в зависимости от параметров инфраструктуры и качества обслуживания, также является важным этапом исследования.

Результат проведенного анализа потребовал формализации данных задач, а также алгоритмизации их решения с учетом особенностей, отраженных на рис. 1.

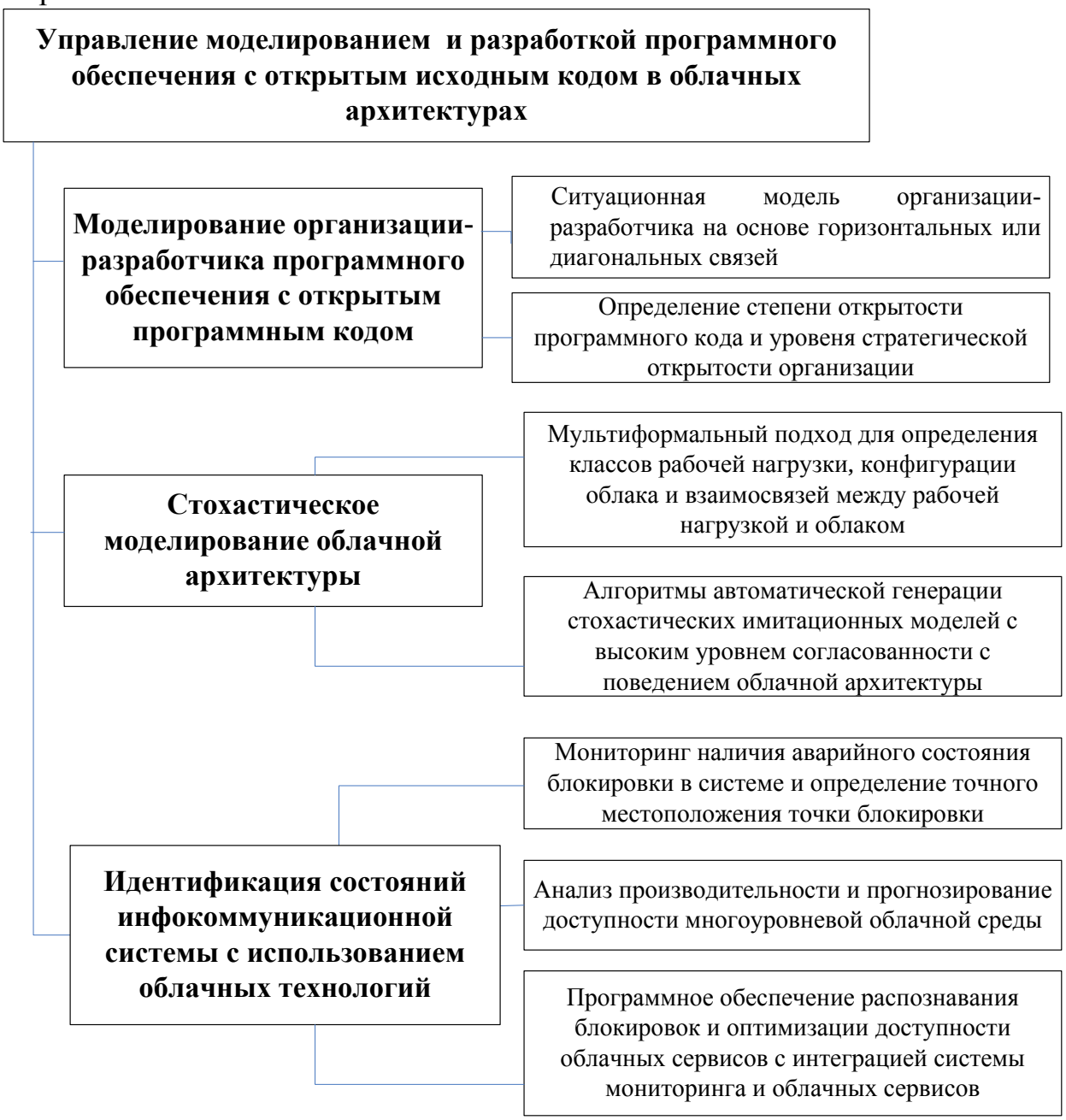


Рис. 1. Дизайн исследования

Сформулирована цель и задачи исследования.

Вторая глава посвящена разработке ситуационной модели анализа организации-разработчика программного обеспечения с открытым программным кодом, обеспечивающей определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

Программные экосистемы определяются как совокупность субъектов, функционирующих как единое целое и взаимодействующих на общем рынке программного обеспечения и услуг, а также отношений между ними. Эти отношения поддерживаются общей технологической платформой или рынком и осуществляются посредством обмена информацией, ресурсами и артефактами. Существует несколько факторов, побуждающих к созданию экосистемы программного обеспечения, таких как способность более гибко реагировать на меняющиеся требования к различные специализированные решения, основанные на платформе, или более быстрое внедрение, чем у конкурентов. Выживание и эффективность организаций в экосистеме взаимозависимы.

Оперирующая система определяется как организация, производящая программное обеспечение и предоставляющая доступ к своим процессам по крайней мере одному из типов участников, которые присутствуют в ее программной экосистеме. Оперирующие системы могут быть любого типа, от независимых поставщиков программного обеспечения до организаций с открытым исходным кодом.

Ситуационная модель направлена на установление того, насколько открытым или закрытым является SPO. Смысл существования модели заключается в трех аспектах. Во-первых, необъективные решения о снабжении принимаются на основе предвзятой оценки открытости, что не позволяет выиграть лучшему продукту или поставщику. Во-вторых, SPO не знают о различных существующих вариантах открытости. В третьих, эта модель направлена на то, чтобы сместить акцент с “открытого исходного кода” и показать, что открытость - это нечто большее, чем просто лицензия на открытое программное обеспечение.

Модель оперирующей системы создана в двух измерениях, одним из которых является измерение практик SPO и измерение уровня управления. В измерении уровня управления есть три уровня: стратегический, тактический и оперативный (от долгосрочного до краткосрочного).

Итоговая ситуационная модель организации-разработчика программного обеспечения с открытым программным кодом, обеспечивающая определение степени открытости программного кода и уровень стратегической открытости организации, представлена на рис. 2.

Первый тип связей - это горизонтальные связи "сверху вниз", в которых один из вариантов открытости от другого варианта открытости на более высоком уровне управления.

Второй тип связей назовем диагональными. Они, как правило, идут сверху вниз слева направо и показывают, что варианты стратегической открытости влияют не только на их собственную практику SPO, но и на других.

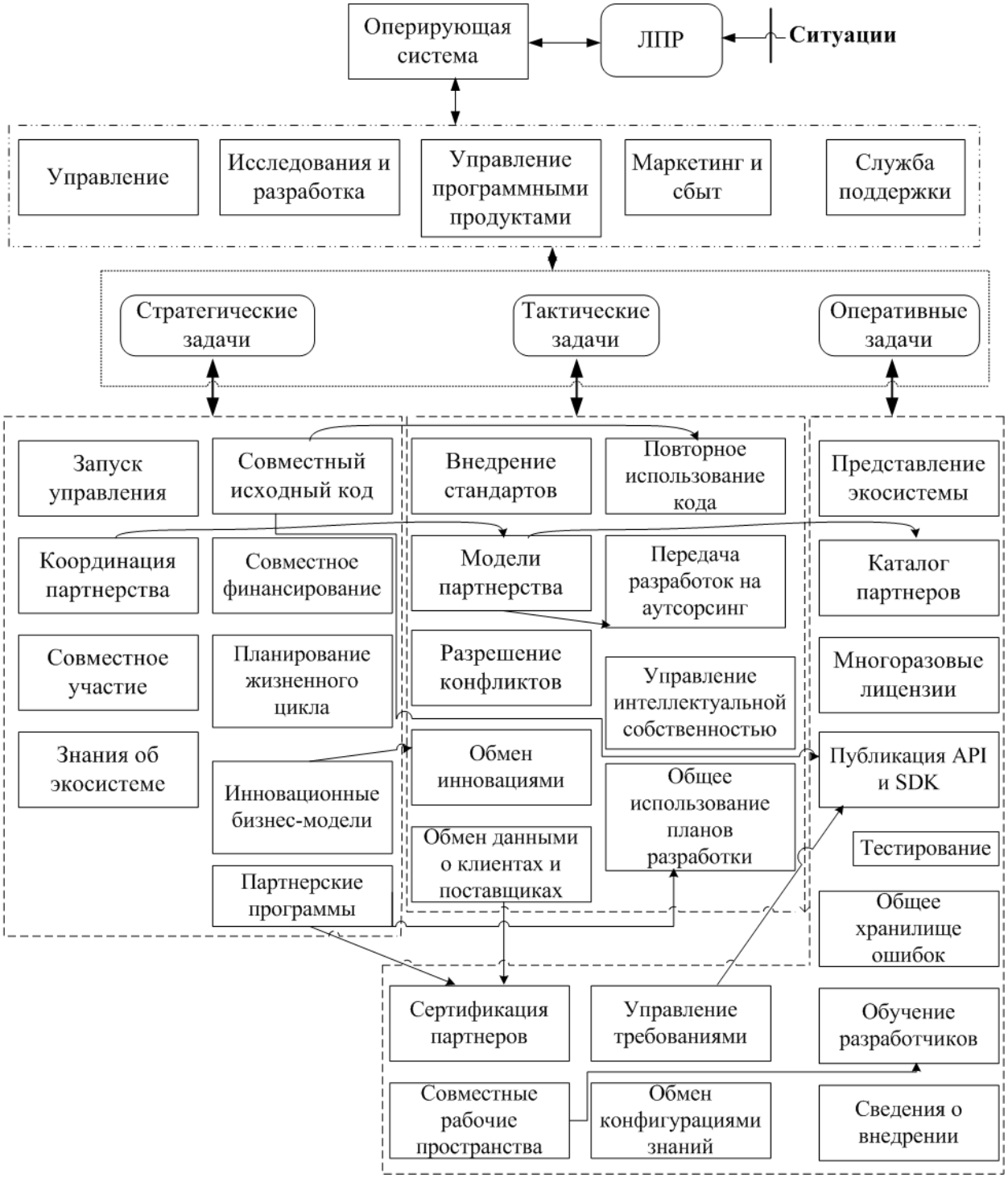


Рис. 2. Структура ситуационной модели анализа организации-разработчика программного обеспечения с открытым программным кодом и примерами связей

Таким образом, в работе предложена ситуационная модель анализа организации-разработчика программного обеспечения с открытым программным кодом, отличающаяся табличным учетом каждого варианта открытости и учетом специфики предметной области продукта на основе горизонтальных или диагональных связей и обеспечивающая определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

Третья глава посвящена разработке модификации технологии стохастического моделирования облачной архитектуры, позволяющей автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с поведением облачной архитектуры.

Рассмотрим рис. 3, на котором представлена эволюция модели облачной архитектуры на протяжении нескольких итераций. Технология направлена на разработку облачных приложений с учетом проблем с производительностью с самого начала цикла проектирования.

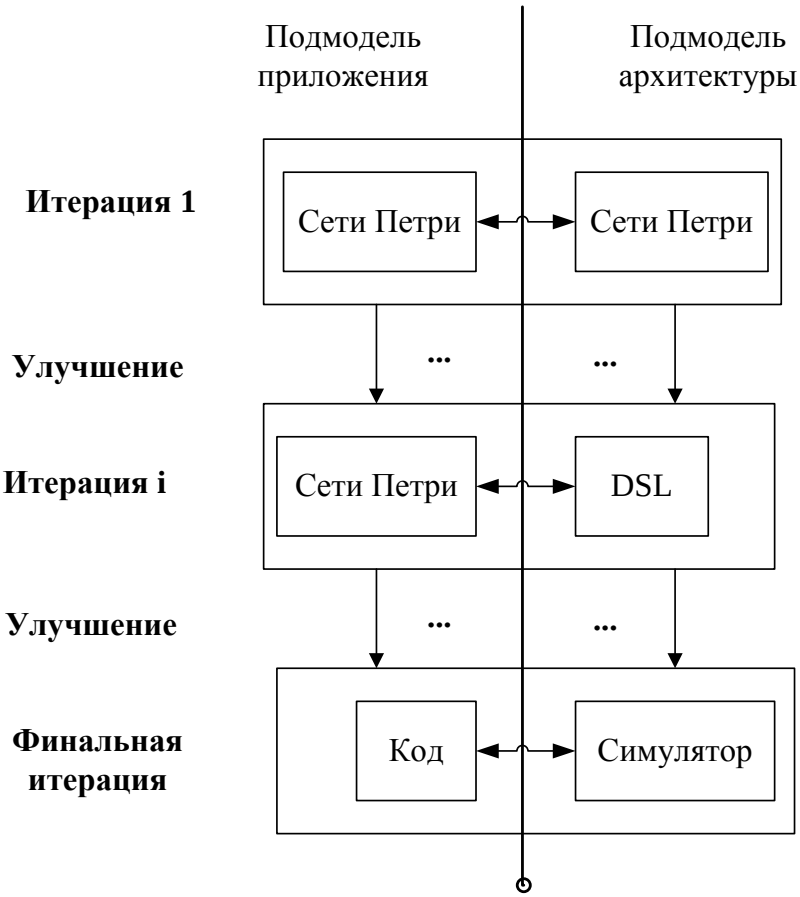


Рис. 3. Информационное взаимодействие модулей моделирования в рамках нескольких итераций

Подход основан на процессе моделирования, при котором разрабатываются модели, состоящие из двух (или двух наборов) подмоделей: первая подмодель представляет приложение, вторая - используемую облачную инфраструктуру. В то время как первое представляет собой нечто, что полностью контролируется разработчиком, о втором ничего не известно с ранних этапов процесса проектирования, и оно становится частично известным, когда выбирается окончательная облачная платформа, поскольку физические детали реализации в любом случае недоступны. На первом этапе обе подмодели могут быть смоделированы с помощью аналитических методов, таких как сети Петри или более специфичный язык, адаптированный к конкретной предметной области, для получения первого приближения оценки производительности и поддержки выбора дизайна. Впоследствии подмодели могут быть доработаны путем добавления деталей: об окончательной реализации

приложения, используя полные знания о нем; о гипотетической целевой архитектуре для облачной инфраструктуры, чтобы экспериментировать с различными конфигурациями.

Следовательно, вся модель разрабатывается до тех пор, пока приложение не будет закодировано и запущено в имитационной облачной среде для окончательной оценки.

Определены два формализмы и их состав, включая все необходимые элементы для создания базовой модели CloudSim. Формализмы описываются элементами и ребрами, каждое из которых обозначается набором свойств (учитывающих параметры, внутреннее состояние и переменные, необходимые для оценки производительности) и поведение (описывающих, как элементы и ребра взаимодействуют для определения общей семантики модели), описывающих формализм и которые могут быть использованы чтобы построить график, представляющий модель. Ребра связывают два элемента и ведут себя соответственно определенной семантике.

Вышеупомянутые формализмы называются, соответственно, GSPN (Обобщенные стохастические сети Петри) и новый DSL, названный CDL (Cloud(Sim) Description Language).

Формализм GSPN является расширением известного формализма SPN. На рис. 4 все элементы и ребра, описывающие GSPN, показаны в правом поле: позиция, временной переход и немедленный переход - это элементы, в то время как ребро и задержка (не используемые в данном примере) представляют ребра.

Рис. 4. Элементы GSPN, CDL и формализмы композиции: 1 – центр обработки данных; 2 – брокер; 3 – хост; 4 – виртуальная машина; 5 – память; 6 – cloudlet; 7 - ребро планирования пуска; 8 – ребро соединения; 9 – ребро рабочей нагрузки

Позиция обладает свойством маркировки, описывающим количество токенов, включенных в данный момент. Временной переход имеет свойство rate, определяющее скорость его срабатывания; мгновенный переход имеет свойство веса, описывающее его относительный вес (в случае, если два мгновенных перехода активированы с помощью отметки общего исходного места,

используемой для стохастического определения того, какой из них выстрелит из двух). Ребро обладает свойством множественности, определяющим, сколько токенов требуется в исходном месте для обеспечения перехода в пункт назначения или создается в месте назначения при запуске перехода. Наконец, задержка обладает свойством множественности, определяющим, сколько токенов в исходном месте необходимо для предотвращения запуска целевого перехода.

Элементы формализма CDL показаны слева на рис. 4. Элементы были выбраны таким образом, чтобы абстрактно представлять все основные концепции, необходимые для создания модели CloudSim.

Общая идея состоит в том, чтобы описать:

1) как организован центр обработки данных с точки зрения его физических и виртуальных машин с помощью подхода, основанного на композиции,

2) визуально распределить рабочие нагрузки с желаемыми характеристиками для брокера, работающего в центре обработки данных.

Формализм композиции определяет только три граничных элемента, соединяющих элементы, принадлежащие двум другим формализмам. Таким образом, они называются гибридными ребрами, или, в более общем смысле, гибридными элементами в терминологии синтеза. Запрос на выполнение arc может подключать только переход GSPN к брокеру, и его семантика соответствует разрешению создания экземпляра рабочей нагрузки и попытке ее выполнения в центре обработки данных. Успешное выполнение arc и неудачное выполнение arc подключают брокера к месту GSPN и выдают токен в этом месте, если выполнение было завершено без проблем или если выполнение было прекращено из-за проблемы.

Пример модели сценария пограничных вычислений приведен на рис. 5.

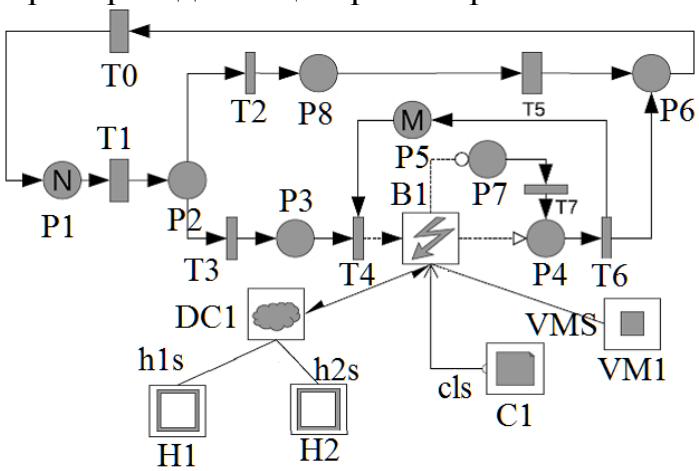


Рис. 5. Модель сценария пограничных вычислений: узлы h1s и h2s с параметрами H1и H2;

На рис. 6 показано среднее время отклика для выполнения одного cloudlet и для всего рабочего процесса (на рисунке обозначено как ”Broker R.T.”). Первый набор показателей получен путем усреднения информации, возвращаемой Cloudsim, описывающей время выполнения каждого облачного пакета. Второй определяется с применением закона Литтла к компоненту

сети Петри, определяющему выполнение одного задания:

$$R_{Broker} = \frac{N - E[P1]}{XT1} \qquad (3.1)$$

где E[P1] и XT1 обозначают соответственно среднюю отметку места P1 и пропускную способность перехода T1.

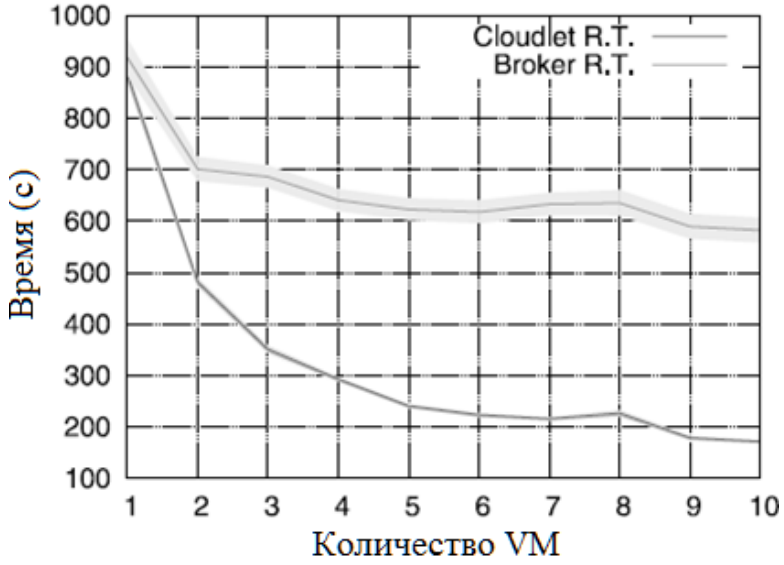


Рис. 6. Среднее время отклика при изменении количества виртуальных машин для каждого брокера

На рис. 7 показан процесс, необходимый для создания симулятора для сценария, описанного в модели. Мультиформальная модель анализируется соответственно с точки зрения

1) описания PN,

2) структуры CDL,

3) гибридных элементов, соединяющих описание и структуру.

На рис. 8 показан общий процесс моделирования с помощью диаграммы последовательности UML.

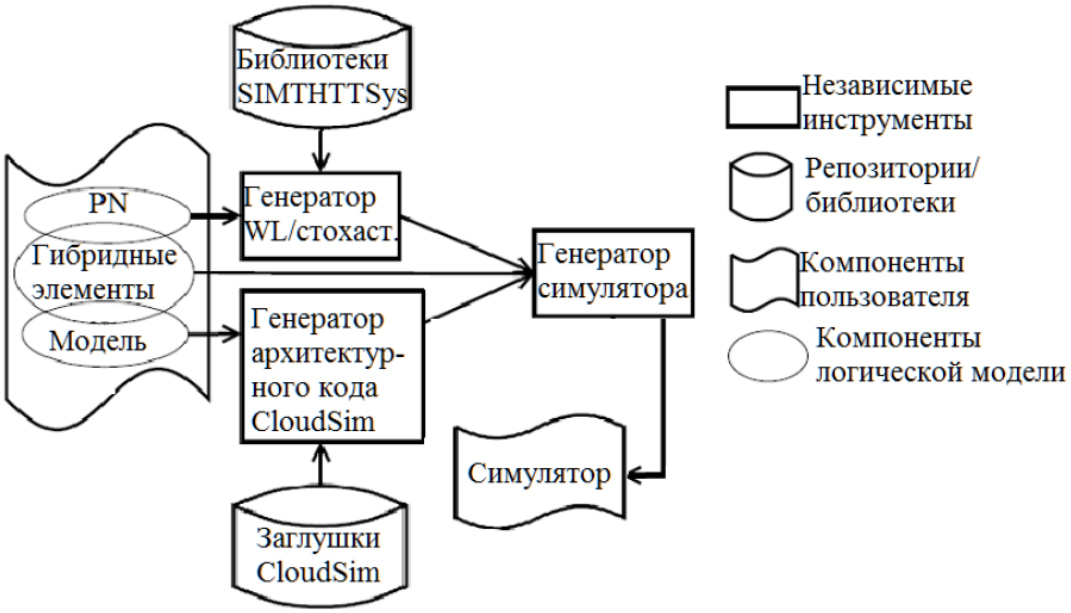


Рис. 7. Процесс генерации симулятора

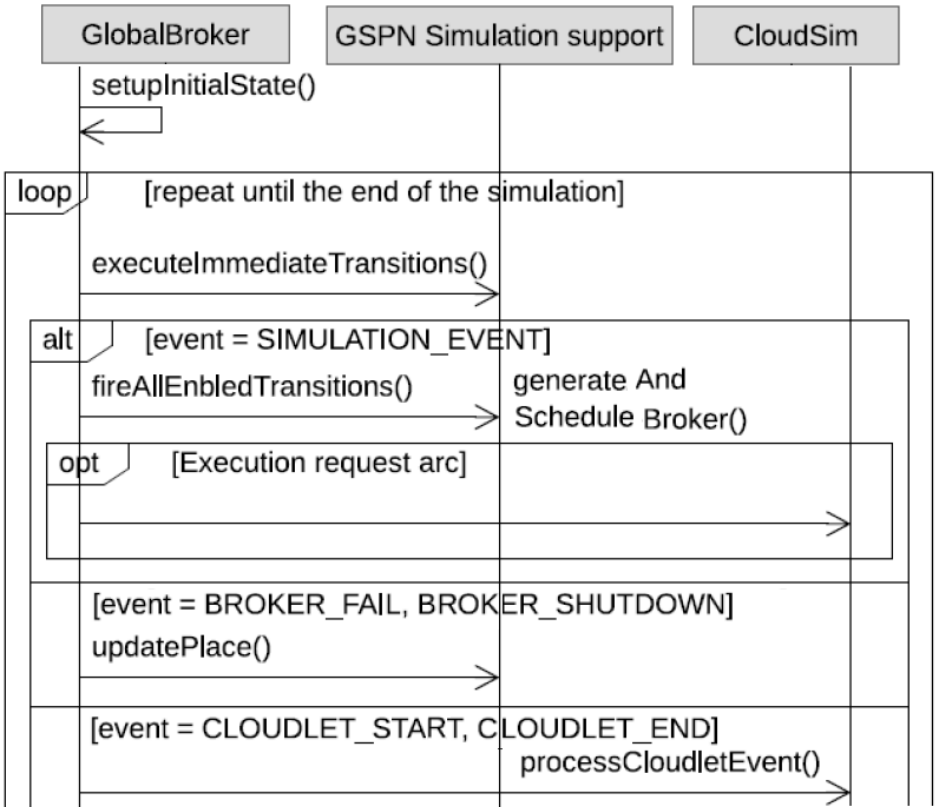


Рис. 8. Фрагмент диаграммы последовательности выполнения процесса моделирования

Подход основан на процессе моделирования, при котором разрабатываются модели, состоящие из двух (или двух наборов) подмоделей: первая подмодель представляет приложение, вторая - используемую облачную инфраструктуру. В то время как первое представляет собой нечто, что полностью контролируется разработчиком, о втором ничего не известно с ранних этапов процесса проектирования, и оно становится частично известным, когда выбирается окончательная облачная платформа.

Таким образом, в работе предложена модификация технологии стохастического моделирования облачной архитектуры, отличающаяся использованием мультиформального подхода для определения классов рабочей нагрузки, конфигурации облака и взаимосвязей между рабочей нагрузкой и облаком в сценариях мультиоблачных/гибридных облачных/пограничных вычислений, позволяющая автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с поведением облачной архитектуры.

В главе 4 представлены модели и инструменты облачного управления доступностью и производительностью инфокоммуникационной подсистемы.

Для решения проблем, связанных с тем, что традиционный метод имеет длительное время отклика на мониторинг перегрузки сети связи, а эффект обнаружения не идеален, предлагается метод мониторинга в реальном времени, основанный на облачных вычислениях для блокировки сети связи. Во-первых, устанавливается точка мониторинга сети связи, и приемник заверша-

ет процесс сбора коммуникационных данных. На основе собранных данных выполняется постоянный расчет трафика для определения наличия состояния блокировки в канале сети связи и определения точного местоположения точки блокировки. Таким образом, информация генерирует тревожное сообщение для получения результатов мониторинга. Экспериментально проанализированы время работы в режиме реального времени и точность метода мониторинга. Установлено, что метод мониторинга позволяет контролировать время задержки в пределах 0,2 с, а частота ошибок мониторинга является низкой.

Несколько устройств сетевого мониторинга могут быть разделены на несколько областей сетевого мониторинга. Одно устройство сетевого мониторинга может логически принадлежать нескольким областям сетевого мониторинга, то есть нескольким службам мониторинга производительности сети для нескольких пользователей, что позволяет сократить количество устройств и снизить затраты на проектирование. Это требует разумного расположения точек сетевого мониторинга и рационального разделения зон сетевого мониторинга и управления ими. Чтобы полностью реализовать функцию точки мониторинга, ее структура разделена на часть локальной вычислительной сети, серверную систему, различные рабочие станции, сервер терминалов, преобразователь протоколов и удаленное сетевое устройство. Структура типичной точки мониторинга показана на рис. 9.

Трафик данных, сети связи рассчитывается в режиме реального времени, и вычислительная структура в режиме реального времени сравнивается с максимальным объемом памяти и максимальной нагрузкой на пропускную способность, и проверяется, может ли трафик данных плавно достигать указанного объема памяти через полосу пропускания для хранения данных.

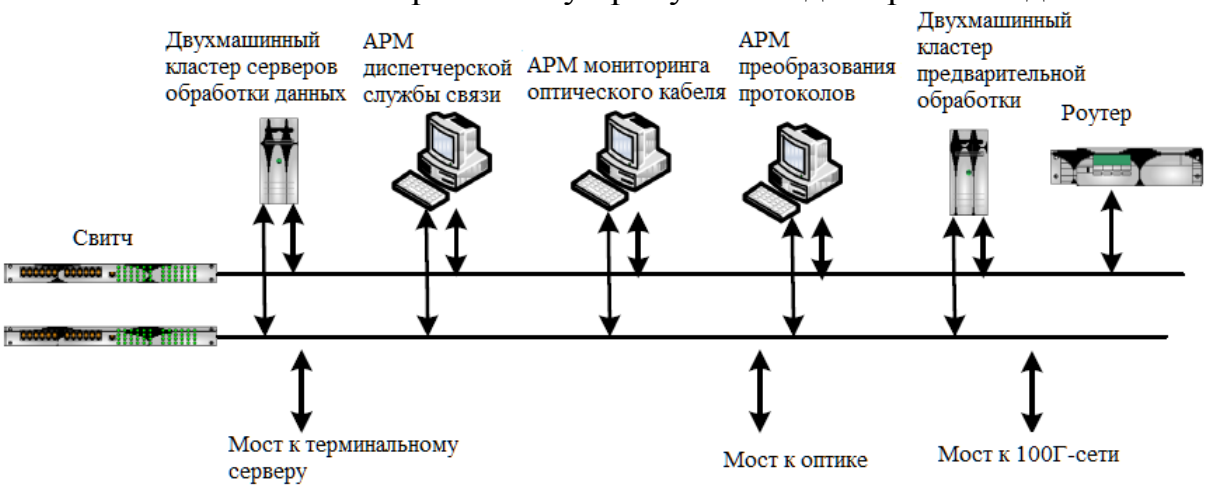


Рис. 9. Типовая схема структуры точки мониторинга

Расчетная формула для расчета потока данных в сети связи в режиме реального времени выглядит следующим образом:

$$Q = \min\left\{\sum_{k\in I_l}\frac{F_l}{\sum\limits_{k\in I_l}P_{lk}y_{lk}-F_l}+G\sum_{\substack{l\in L\\k\in I_l}}\left(S_{lk}y_{lk}+d_i m_{lk}\right)+V\sum_{\substack{l\in L\\k\in I_l}}\left(C_{lk}F_{ly_{lk}}\right)\right\} \qquad (4.1)$$

где Q – общий трафик данных; F_l - локальный трафик данных по каналу инфокоммуникационной сети (ИКС); I_l - набор индикаторов модели потенциального соединения для первой связи; P_{lk} - индекс модели первого звена, выбранного в качестве пропускной способности связи k; S_{lk} - возможный набор маршрутов для узла первой связи; C_{lk} - коэффициент линейной переменной при выборе индекса модели первого звена; d_i – длина первой связи; m_{lk} - соединитель; I_l - скорость поступления пакетов; G, V – фиксированный весовой коэффициент; L – множество всех связей КС.

Ограничения:

$$\sum_{k \in S_p} x_p = 1 \; \forall p \in \Pi \tag{4.2}$$

$$\sum_{k \in I_l} y_{lk} = 1 \; \forall l \in L \tag{4.3}$$

где P - набор всех пар узлов связи в сети; x_p - переменная оптимизации, равна 1, когда маршрут выбран, иначе 0; y_{lk} - переменная оптимизации. Когда выбран индекс модели первого канала «ask», значение равно 1, в противном случае оно равно 0; может быть получен коммуникационный трафик в реальном времени по определенному каналу, а также могут быть получены максимальный предел и пропускная способность дискового пространства.

Когда система мониторинга трафика фиксирует передаваемый кадр Ethernet, ей необходимо сначала проанализировать пакет данных, а затем извлечь соответствующие данные и сохранить результат извлечения в БД, что удобно для анализа аномального сетевого трафика в режиме реального времени. Чтобы обеспечить точность системного перехвата, необходимо сначала прочитать конфигурационный файл; затем установить соединение с БД, зарегистрировать источник данных ODBC, использовать функцию openDatabase() в файле скрипта Python для подключения к БД; настроить драйвер перехвата для создания различных типов, создать таймеры и потоки, мониторинг аномального трафика в режиме реального времени с использованием таймеров; наконец, создать сервер мониторинга в режиме реального времени и вызвать функцию Bind(), чтобы получить IP-адрес сервера мониторинга из файла конфигурации, тем самым завершив процесс перехвата пакетов. Пакет данных о ненормальном трафике может быть перехвачен в режиме реального времени, и может быть добавлена функция отображения мониторинга в режиме реального времени, чтобы пользователь мог просматривать результат захвата пакета в режиме реального времени, а затем находить аномальный узел.

Чтобы проверить целесообразность применения метода мониторинга блокировки ИКС в режиме реального времени, были проведены эксперименты. Сглаженные результаты сравнения показаны на рис. 10.

В качестве экспериментальных объектов выбирались 50 наборов данных за определенный период времени, и данные о нормальном состоянии получаются в соответствии с историческими записями за предыдущие периоды, после чего выполняется стандартизированная обработка. Использовался симулятор Mininet для создания топологии сети связи на виртуальной машине и

подключения к удаленному контроллеру для реализации построения сетевой среды связи. В качестве экспериментальных показателей были взяты ошибка мониторинга в режиме реального времени и время задержки связи в ИКС. Традиционный метод мониторинга сравнивается с мониторингом трафика в сети связи в режиме реального времени. Алгоритм идентификации состояния ИКС представлен на рис. 111.

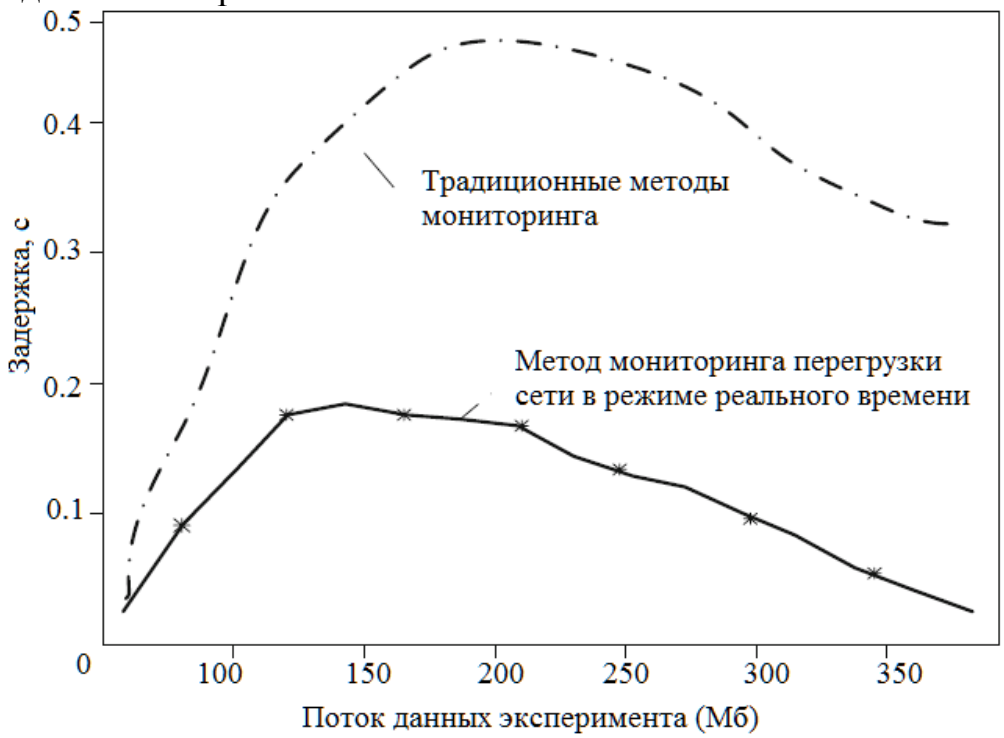


Рис. 10. Сглаженные результаты сравнения времени задержки связи

Рис. 11. Алгоритм идентификации состояния КС

Таким образом, представлен алгоритм идентификации состояния инфокоммуникационной системы на основе облачных вычислений, отличающийся событийным мониторингом ситуаций блокировки системы и обеспечивающий расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки.

Доступность является важным качеством облачной системы. Доступность может гарантировать качество обслуживания (QoS) в облачных системах. В работе рассматривается проблема традиционного моделирования производительности, а также предлагается решение для анализа производительности в многоуровневом облаке. В этом моделировании среднее время жизни отдельных компонентов объединяется в целостную многоуровневую систему для расчета доступности. Моделирование выполняется с двух разных точек зрения, например, на основе инфраструктуры и приложений. При моделировании на основе инфраструктуры рассматривается набор компонентов, таких как виртуальная машина (VM) и хост-машина (HM). Основные компоненты VM и HM включают в себя такие их подкомпоненты, как загрузка процессора, памяти, диска, передача данных и сетевых байтов. В предлагаемой модели рассматриваются эти конечные компоненты, которые могут повлиять на производительность многоуровневой облачной среды. В модели, основанной на приложении, учитывается набор показателей, таких как метрика хоста, пропускная способность, использование ресурсов и виртуальные показатели. После оценки срока службы этих компонентов вычисляется среднее время наработки на отказ (MTTF). Доступность многоуровневой облачной системы определяется по MTTF виртуальной машины и HM. На основе этой доступности выполняется анализ производительности. Доступность оценивается путем объединения набора важных параметров, чтобы приблизительное предсказание доступности было более точным в этой модели по сравнению с известными моделями доступности. Предлагаемое моделирование оценивается путем реализации модели в инструменте SHARPE. Результатом этой работы является разработка модели производительности, которая позволяет измерить любую облачную систему на основе ее доступности и проанализировать качество предоставляемых ею услуг.

Частота отказов активного компонента λ определяется на основе мониторинга набора соответствующих компонентов. Пусть A_i ($1 \le i \le M$) обозначает время жизни активного компонента V, пусть B_i ($1 \le i \le M$) обозначает время жизни активного компонента H и пусть D_j ($1 \le j \le S$) обозначает время жизни неактивного компонента. Тогда время жизни многоуровневой облачной системы $L_T(n|M,C)$ определяется как

$$\begin{aligned} &L_T(n \mid M,S) = L_T(n \mid V,H,S) = \\ &= \min\left(A_1, A_2, ..., A_m; B_1, B_2, ..., B_m; D_1, D_2, ..., D_S\right) + \\ &+ L_T(n \mid M, S-1) = \\ &= W_F(H,V,S) + L_T(n \mid H,V,S-1) \end{aligned} \qquad (4.5)$$

$W_F(H,V,S)$ − это время, в течение которого в системе происходит сбой между H+V+S компонентами системы и неисправные компоненты удаляют-

ся, облачная система имеет M активных и S-1 деактивированных компонентов. Для M+S−1 нет времени на старение.

$$L_T(n \mid H,V,0) = L(n \mid H,0) + \sum_{j=1}^{S} W_F(H,j) + L(n \mid V,0) + \sum_{j=1}^{S} W_F(V,j) \quad (4.6)$$

Здесь LT(n|H,V,0)=LT(n| H,V) - это время жизни системы вне H&V и, следовательно, статистика l-го порядка с l=H+V−n+1. Для определения вероятности жизни используем гипоэкспоненциальное распределение. Распределение L_T(n|H,V,0) является гипоэкспоненциальным с параметрами (H+V)λ, ((H+V)-1)λ,.... nλ. Это может быть индивидуально распределено L(n|H,0) как Hλ, (H−1)λ,.... nλ и L(n|V,0) как Vλ, (V−1)λ, ... nλ. У активной системы есть частота отказов λ в зависимости от загрузки процессора, диска и задержки. Для расчета надежности системы оценивается частота отказов и частота ремонтов. Таким образом, L_T(n|M,S) имеет (H+V+S–(n+1)) ступенчатое гипоэкспоненциальное распределение с параметром (H+V)λ+Sμ, (H+V)λ+(S−1) μ, …, (H+V) λ+μ, (H+V)λ, ((H+V)-1) λ,…, nλ.

Пусть R_L[n|M,S](t) - надежность всего многоуровневого облака:

$$R_L\left[n \mid H,V,S\right](t) = \sum_{j=1}^{S} a_j e^{-((H+V)\lambda + j\mu)t} + \sum_{j=1}^{H+V} b_j e^{-(\lambda j)t} \quad (4.7)$$

$$\begin{aligned} R_L\left[n \mid M,S\right](t) &= \sum_{j=1}^{S} a_j e^{-(M\lambda + j\mu)t} + \sum_{j=n}^{H+V} b_j e^{-(\lambda j)t} = \\ &= \sum_{j=1}^{S} a_j e^{-(H\lambda + j\mu)t} + \sum_{j=1}^{S} a_j e^{-(V\lambda + j\mu)t} + \sum_{j=n}^{H} b_j e^{-(j\lambda)t} + \sum_{j=n}^{V} b_j e^{-(j\lambda)t} \end{aligned} \quad (4.8)$$

где a_j - время жизни неактивной системы с постоянной частотой отказов μ, b_j - время жизни активной системы с частотой отказов λ.

Из приведенного выше уравнения (4.8) время жизни отдельных компонентов VM и HM можно определить как

$$R_L\left[n \mid H,V,S\right](t) = \sum_{j=n}^{H+V} b_j e^{-(\lambda j)t} \quad (4.9)$$

Доступность измеряется с помощью инструмента SHARPE. Символьно-иерархическая автоматизированная оценка надежности и производительности (SHARPE) - это инструмент моделирования, который сочетает в себе гибкость марковской модели и эффективность комбинаторной модели. В эксперименте SHARPE используется для разработки высокоуровневой архитектуры многоуровневого облака и применения предложенного уравнения к разрабатываемому облаку.

На рис. 12 показано сравнение точности оценки доступности между предложенной моделью доступности и стандартной.

Результат показывает, что предложенная модель является более точной по сравнению со стандартными моделями. Она предсказывает значение доступности с точностью приблизительно 99%, на 8.3% точнее стандартных.

Таким образом, создана модель анализа производительности и прогнозирования доступности многоуровневой облачной среды, отли-

чающаяся периодическим расчетом времени жизни облачной системы и обеспечивающая учет времени жизни основных компонентов и оценку доступности облачной среды. Модель предсказывает значение доступности на 8.3% точнее стандартных.

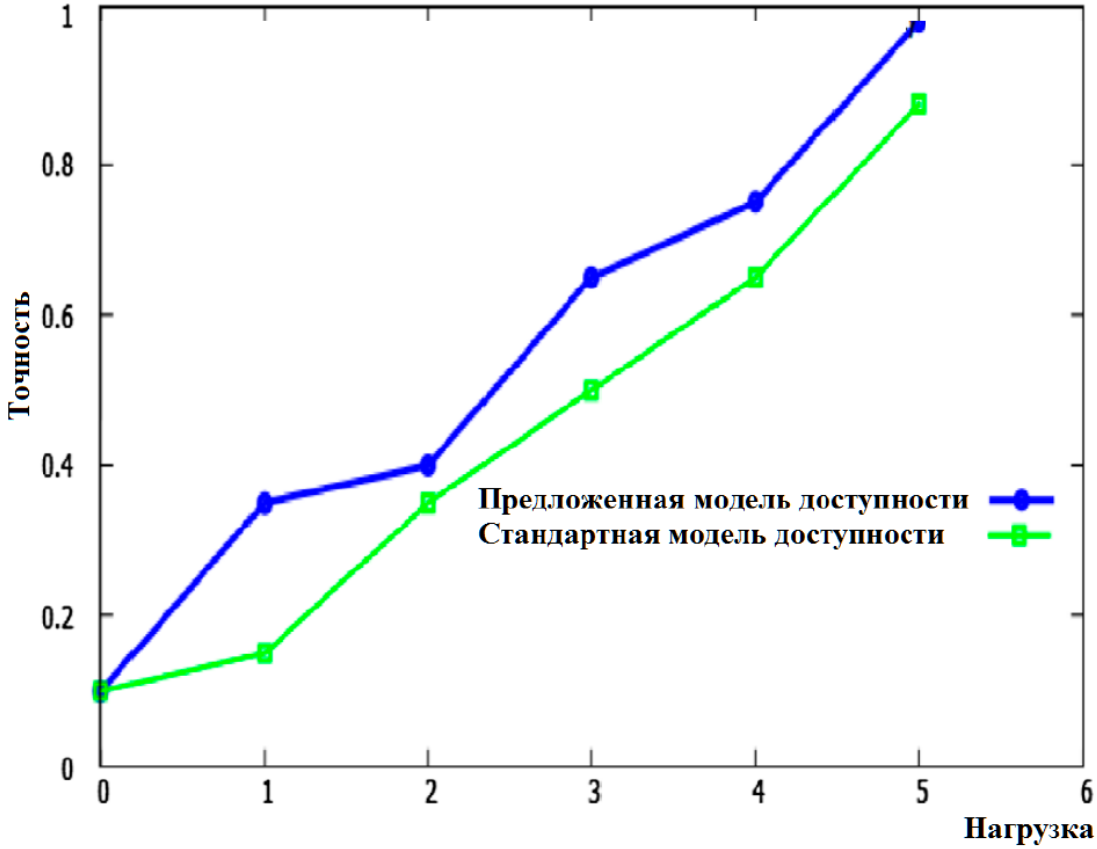


Рис. 12. Сравнение точности

Разработана структура программного прототипа системы распознавания блокировок и оптимизации доступности облачных сервисов (рис. 13). Система мониторинга активно через датчики и иные инструменты решает задачу идентификации отказа.

Далее отрабатывает модуль распознавания аномальной блокировки. В такой ситуации система автоматически или через ЛПР осуществляет реконфигурацию коммуникационной сети.

Затем проводится расчет параметров системы в реальном масштабе времени. Данные с выхода системы мониторинга также попадают на вход модуля оценки параметров новой инфраструктуры. Отрабатывает блок анализа доступности инфраструктуры и следом модуль оценки производительности облака.

Данные поступают на вход модуля оценки качества обслуживания. В зависимости от автоматического или автоматизированного (с участием ЛПР) оценивания качества возможен возврат к реконфигурации.

Элементы программной реализации прошли государственную регистрацию в ФИПС.

Рис. 13. Структура программного прототипа системы распознавания блокировок и оптимизации доступности облачных сервисов

Таким образом, создана структура программного обеспечения распознавания блокировок и оптимизации доступности облачных сервисов, отличающаяся интеграцией системы мониторинга и облачных сервисов и обеспечивающая реконфигурацию инфокоммуникационной системы в зависимости от параметров инфраструктуры и качества обслуживания.

Заключение

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Создана ситуационная модель анализа организации-разработчика программного обеспечения с открытым программным кодом, обеспечивающая определение степени открытости программного кода и уровень стратегической открытости организации в экосистеме жизненного цикла.

2. Предложена модификация технологии стохастического моделирования облачной архитектуры, позволяющая автоматически генерировать стохастические имитационные модели с высоким уровнем согласованности с

поведением облачной архитектуры

3. Разработан алгоритм идентификации состояния инфокоммуникационной системы на основе облачных вычислений, обеспечивающий расчет трафика для определения наличия состояния блокировки в системе и определения точного местоположения точки блокировки

4. Предложена модель анализа производительности и прогнозирования доступности многоуровневой облачной среды, обеспечивающая учет времени жизни основных компонентов и оценку доступности облачной среды. Модель предсказывает значение доступности на 8.3% точнее стандартных.

5. Разработана структура программного обеспечения распознавания блокировок и оптимизации доступности облачных сервисов, обеспечивающая реконфигурацию инфокоммуникационной системы в зависимости от параметров инфраструктуры и качества обслуживания.

6. Элементы программного обеспечения зарегистрированы в ФИПС.

Рекомендации и перспективы дальнейшей разработки темы

1. Результаты исследования рекомендуются к применению в задачах исследования и разработки программного обеспечения с открытым исходным кодом в облачных архитектурах.

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

Основные результаты диссертации опубликованы в следующих работах:

Публикации в изданиях списка ВАК

1. Атласов И.В., Горшков А.В., Каперко А.Ф., Амоа Куадио-кан Армел Жеафруа, Коптелова А.С. Информационное представление метода и модели кэширования на основе использованием временных меток// Системы управления и информационные технологии, №2(96), 2024. - С. 46-50.

2. Амоа Куадио-кан Армел Жеафруа, Баталов Д.И., Балдин А.В., Мутина Е.И. Моделирование облачных систем с использованием формализмов и программных интерфейсов CloudSim// Системы управления и информационные технологии, №3(97), 2024. С. 55-59.

3. 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

Публикация в издании, индексируемом в WoS

4. Kravets O.Ja., Aksenov I.A., Redkin Yu.V., Rahman P.A., Mutin D.I., Amoa Kouadio-kan Armel Geoffroy, Ermolova M.A. Tools for designing complex software systems based on special linguistic constructions and algorithms for their processing// International Journal on Information Technologies and Security,

vol.16, no.4, 2024, pp. 15-24. https://doi.org/10.59035/XOIF5262. WOS:001369038100002.

Свидетельство о государственной регистрации программы для ЭВМ

5. 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

6. Амоа К.К.А.Ж. Многоагентный подход к построению распределенной архитектуры веб-приложения// Современная наука: актуальные проблемы теории и практики. Серия: Естественные и Технические Науки. - 2022. - №10. - С. 47-51.

7. Амоа Куадио-кан Армел Жеафруа. Оперирующая система в экосистеме организации-разработчика программного обеспечения// Информационные технологии моделирования и управления, №2(136), 2024. – С. 88-100.

8. Амоа Куадио-кан Армел Жеафруа, Рындин Н.А. Модель OSE в экосистеме организации-разработчика программного обеспечения// Экономика и менеджмент систем управления, №2(52), 2024. – С. 39-48.

9. Амоа Куадио-Кан Армел Жеафруа. Проблемы и особенности использования формализмов при исследовании облачных систем// Наука и технологии: перспективы развития и применения: сб. статей VII Междунар. НПК. - Петрозаводск: МЦНП «НОВАЯ НАУКА», 2024. - С. 59-64.

10. Амоа Куадио-кан Армел Жеафруа. Программные решения для имитационного моделирования сложных систем с участием облачных технологий// Информационные технологии моделирования и управления, №3(137), 2024. – С. 222-238.

11. Амоа Куадио-кан Армел Жеафруа. Исследование пограничных вычислений с использованием Cloudsim// Сб. тр. VI Всеросс. НПК «Информационные технологии в экономике и управлении». – Махачкала, 2024. С. 59-64.

12. Amoa Kouadio-kan Armel Geoffroy, Ryndin N.A. Research of boundary computing based on modeling of cloud systems using formalisms and software interfaces CloudSim// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2025'AS): Proc. of the XXX-th Int. Open Science Conf. - Yelm, WA, USA: Science Book Publishing House, 2025. – pp. 140-148.

Подписано в печать 25.04.25
Формат 60x84/16. Бумага для множительных аппаратов.
Усл. печ. л. 1,0. Тираж 80 экз. Заказ №199.
ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14