

На правах рукописи



Азиз Аммар Имад Азиз

**СПЕЦИАЛЬНОЕ МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ ПРОЦЕССА БЕЗОПАСНОГО УПРАВЛЕНИЯ  
РЕПЛИКАЦИЯМИ В МАСШТАБИРУЕМЫХ СУБД**

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

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата технических наук

Воронеж – 2023

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего образования «Воронежский государственный университет»

Научный руководитель: **Бондаренко Юлия Валентиновна**,  
доктор технических наук, доцент

Официальные оппоненты: **Ланкин Олег Викторович**,  
доктор технических наук, доцент,  
Воронежский институт правительственной связи  
(филиал) Академии Федеральной службы охраны  
Российской Федерации, начальник учебного от-  
дела

**Преображенский Юрий Петрович**,  
кандидат технических наук, доцент,  
Воронежский институт высоких технологий – ав-  
тономная некоммерческая образовательная орга-  
низация высшего образования, проректор по ин-  
формационным технологиям

Ведущая организация: Федеральное государственное бюджетное обра-  
зовательное учреждение высшего образования  
«Тамбовский государственный технический уни-  
верситет»

Защита состоится «20» октября 2023 года в 14 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 [www.cchgeu.ru](http://www.cchgeu.ru).

Автореферат разослан «3» сентября 2023 г.

Ученый секретарь  
диссертационного совета



Гусев Константин Юрьевич

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы.** Объем данных, генерируемых мобильными гаджетами, датчиками и другими вычислительными устройствами, явился причиной создания концепции больших данных.

В этом контексте СУБД NoSQL стали популярными благодаря большей производительности для управления большими наборами данных с помощью альтернативных моделей данных. Многие службы облачных вычислений используют СУБД NoSQL, но оценка качества обслуживания (QoS) создает дополнительные проблемы для этих систем. Такими исследованиями занимались Вишневский В.М., Денисов А.А., Кравец О.Я., Мелентьев В.А., Bruneo D., Chakraborty S., Mohan C. Интересным является подход, основанный на обобщенных стохастических сетях Петри (GSPN) для обеспечения производительности частных облачных вычислительных сред, которые принимают СУБД NoSQL как систему хранения.

Развитие Интернет вещей (IoT) и популяризация мобильных устройств значительно поспособствовали созданию новой концепции, а именно Больших данных, которая характеризует скорость, объем и разнообразие данных. В этом контексте СУБД NoSQL стали замечательной альтернативой реляционным СУБД для решения проблем с большими данными. СУБД NoSQL обычно обладают лучшей производительностью для обработки больших наборов данных, их легко развернуть как распределенную систему и они более эффективно управляют неструктурированными данными. Кроме того, NoSQL позволяет устанавливать различные уровни согласованности между резервными серверами.

Следует отметить, что многие сервисы облачных вычислений используют NoSQL, но существуют дополнительные проблемы для оценки показателей качества обслуживания (QoS), например, из-за различных рабочих нагрузок и влияния доступности. При этом интересной представляется идея совместной оценки эффективности и доступности (т.е. производительности) частных облачных вычислительных систем, использующих СУБД NoSQL в качестве подсистемы хранения.

В этой связи актуальность исследования определяется необходимостью разработки дополнительных компонентов математического и программного обеспечения процесса безопасного управления репликациями в масштабируемых СУБД на основе реализации соответствующих алгоритмических средств.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный университет» «Суперкомпьютерные технологии, квантовые и распределенные вычисления, большие данные».

**Целью работы** является развитие средств математического и программного обеспечения процесса безопасного управления репликациями в масштабируемых СУБД на основе разработки специального мультиагентного самонастраивающегося алгоритма.

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

1. Разработать модель частных облачных вычислительных систем с NoSQL-подсистемами хранения данных на основе обобщенных стохастических

сетей Петри.

2. Создать алгоритм динамической балансировки системы, основанный на многокритериальном принятии решений при формировании оптимальной матрицы альтернатив.

3. Разработать мультиагентный самонастраивающийся алгоритм репликации, основанный на обучении с подкреплением с использованием монитора метрик, диспетчера обновлений и агентов обучения с подкреплением.

4. Создать сервис-ориентированную архитектуру несвязанной СУБД с модифицированной подсистемой ведения журнала транзакций и механизмом его очистки с нулевой памятью.

5. Разработать архитектуру человеко-машинной системы проектирования базовой структуры БД с интерактивным характером выбора оптимальных характеристик межсистемного взаимодействия.

**Объект исследования:** системы облачных вычислений.

**Предмет исследования:** математические средства улучшения безопасного управления репликациями в масштабируемых СУБД.

**Методы исследования.** При решении поставленных в диссертации задач использовались методы 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 частных облачных вычислительных систем с подсистемами хранения данных на основе NoSQL, отличающаяся использованием обобщенных стохастических сетей Петри и обеспечивающая корректную оценку пропускной способности системы при различных рабочих нагрузках, доступности и простоях.

2. Алгоритм динамической балансировки системы, основанный на многокритериальном принятии решений при получении оптимальной матрицы альтернатив и обеспечивающий повышение производительности и QoS с одновременным определением приоритетов процессов.

3. Мультиагентный самонастраивающийся алгоритм репликации, основанный на обучении с подкреплением с использованием монитора метрик, диспетчера обновлений и агентов обучения с подкреплением, обеспечивающий конфигурирование гибридного промежуточного программного обеспечения репликации без перезапуска СУБД в режиме реального времени.

4. Сервис-ориентированная архитектура несвязанной СУБД, отличающаяся модифицированной подсистемой ведения журнала транзакций и механизмом его очистки с нулевой памятью, обеспечивающая снижение служебного

сетевого трафика и повышение безопасности обработки при проектировании разделения вычислений и данных.

5. Архитектура человеко-машинной системы проектирования базовой структуры БД, отличающаяся интерактивным характером выбора оптимальных характеристик межсистемного взаимодействия и обеспечивающая автоматизацию динамического включения новых подмоделей в структуру БД.

**Теоретическая и практическая значимость исследования** заключается в разработке средств специального математического и программного обеспечения процесса безопасного управления репликациями в масштабируемых СУБД на основе многокритериального принятия решений при получении оптимальной матрицы альтернатив, а также средств информационного и программного обеспечения для экспериментальной оценки качества разработанных алгоритмов.

Теоретические результаты работы могут быть использованы в проектных и научно-исследовательских организациях, занимающихся проектированием платформенно-инвариантных систем управления облачными средами в условиях штатной или нерегламентированной внешней нагрузки.

#### **Положения, выносимые на защиту**

1. Модель частных облачных вычислительных систем с подсистемами хранения данных на основе NoSQL обеспечивает корректную оценку пропускной способности системы при различных рабочих нагрузках, доступности и простоях.

2. Алгоритм динамической балансировки системы обеспечивает повышение производительности и QoS с одновременным определением приоритетов процессов.

3. Мультиагентный самонастраивающийся алгоритм репликации, основанный на обучении с подкреплением, позволяет осуществить конфигурирование гибридного промежуточного программного обеспечения репликации без перезапуска СУБД в режиме реального времени.

4. Сервис-ориентированная архитектура несвязанной СУБД с модифицированной подсистемой ведения журнала транзакций обеспечивает снижение служебного сетевого трафика и повышение безопасности обработки при проектировании разделения вычислений и данных.

5. Архитектура человеко-машинной системы проектирования базовой структуры БД обеспечивает автоматизацию динамического включения новых подмоделей в структуру БД.

**Результаты внедрения.** Основные результаты внедрены в виде программных компонент систем управления транзакциями в АО НПП «РЕЛЭКС» (г. Воронеж), «Al Rawiyah for information technology» company (Багдад, Ирак), а также в учебный процесс Воронежского государственного университета в рамках дисциплин «Базы данных», «Информационная безопасность и защита информации», «Информатизация предприятия», а также в рамках курсового и дипломного проектирования.

**Апробация работы.** Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: Всероссийской науч-

ной конференции «Достижения науки и технологий-ДНиТ-2021» (Красноярск, 2021); Международной научной конференции «Актуальные проблемы прикладной математики, информатики и механики» (Воронеж, 2021); XXVII-th International Open Science Conference «Modern informatization problems in economics and safety (MIP-2022'ES)» (Yelm, WA, USA, 2022), Международной научной конференции «Актуальные проблемы прикладной математики, информатики и механики» (Воронеж, 2021), а также на научных семинарах кафедры математических методов исследования операций ВГУ (2020-2023 гг.).

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

**Публикации.** По результатам диссертационного исследования опубликовано 11 научных работ (4 – без соавторов), в том числе 6 – в изданиях, рекомендованных ВАК РФ (из них 2 – в издании, индексируемом в международной базе данных Scopus, и одно свидетельство о регистрации программы для ЭВМ). В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [5] – модель частных облачных вычислительных систем с подсистемами хранения данных на основе NoSQL; [6, 9] – алгоритм динамической балансировки системы; [7, 8] – мультиагентный самонастраивающийся алгоритм репликации; [10] – сервисориентированная архитектура несвязанной СУБД с модифицированной подсистемой ведения журнала; [11] – информационное и программное обеспечение человеко-машинной системы проектирования базовой структуры БД.

**Структура и объем работы.** Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 194 наименований. Работа изложена на 157 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 разработки математического и программного обеспечения безопасного управления репликациями в масштабируемых СУБД на основе разработки мультиагентных самонастраивающихся алгоритмов репликации и анализируется современное состояние проблемы управления ими.

Производительность баз данных является важной областью исследований, поскольку хранение данных играет фундаментальную роль во многих вычислительных системах. На протяжении многих лет исследователи также уделяли внимание оценке производительности СУБД NoSQL. Общий подход, принятый в NoSQL, заключается в сравнении производительности с учетом различных рабочих нагрузок. Например, в этих работах рассматривается один или несколько серверов (т.е. распределенных систем), и они обычно используют пропускную способность и время отклика в качестве интересующих показате-

лей. Результаты показывают наиболее подходящие СУБД NoSQL для каждой конкретной рабочей нагрузки. Необходима совместная оценка эффективности и доступности (производительности) частных облачных вычислительных систем, использующих СУБД NoSQL в качестве подсистемы хранения.

В настоящее время существует множество СУБД, каждая из которых ориентирована на определенный тип рабочей нагрузки, а именно: оперативная обработка транзакций (OLTP), оперативная аналитическая обработка (OLAP) и даже более новые гибридные рабочие нагрузки по транзакционной и аналитической обработке (HTAP). Несмотря на то, что существуют рекомендации по оценке и настройке СУБД, конфигурации не являются универсальными, и каждый поставщик решает, какие настраиваемые конфигурации следует предоставлять. Кроме того, типы рабочей нагрузки очень различаются, чтобы обеспечить общую конфигурацию в разных системах. Это делается для того, чтобы оптимизация целевой операции OLTP существенно снизила производительность OLAP, и наоборот. Поскольку тип конфигураций, доступных для настройки, увеличивается и проблемы, связанные с данным типом рабочей нагрузки, различны для каждого поставщика, администратор базы данных ответственен за оценку и точную настройку рабочей нагрузки в конкретной СУБД.

Это особенно верно для механизмов репликации и обеспечения отказоустойчивости и высокой доступности, поскольку такие функции обычно тесно связаны с тем, как каждая СУБД обрабатывает данные, особенно в системах OLTP. Системы баз данных обычно обеспечивают отказоустойчивость за счет снижения пропускной способности репликации, когда алгоритмы сложно развертываются как часть ядра СУБД. При сопряжении подсистемы СУБД с обеспечением гарантий отказоустойчивости через внешние системы промежуточного программного обеспечения количество рассматриваемых настраиваемых конфигураций увеличивается.

Отмечено, что повысить производительность и безопасность управления репликациями в масштабируемых СУБД можно путем модификации динамической балансировки системы, основанной на многокритериальном принятии решений при получении оптимальной матрицы альтернатив; а также сервис-ориентированной архитектуры СУБД с модифицированной подсистемой ведения журнала транзакций и механизмом его очистки с нулевой памятью. Потребовались формализация данных задач, а также алгоритмизация их решения с учетом особенностей. Сформулирована цель и задачи исследования.

**Вторая глава** посвящена исследованию производительности облачных вычислительных сред на платформе СУБД NoSQL на основе обобщенных стохастических сетей Петри и многокритериальных оценок.

СУБД NoSQL не используют свойства ACID, но они используют основную концепцию. Без потери общности последнее фокусируется на производительности за счет использования возможной согласованности. В работе используются обобщенные стохастические сети Петри (GSPN), важное расширение сетей Петри (PN), которое позволяет связать экспоненциальное распределение с временными переходами или нулевыми задержками с немедленными переходами. Кроме того, временные переходы могут иметь разные степени параллеле-

лизма: семантика одного сервера (ss) или семантика бесконечного сервера (is). Модель может оценивать производительность системы при различных рабочих нагрузках, доступности и времени простоя. Подход к моделированию основан на моделях строительных блоков с использованием моделей меньшего размера для создания окончательного представления системы. Модель может оценивать производительность системы при различных рабочих нагрузках, доступности и времени простоя. Все переходы используют экспоненциальное распределение и семантику одного сервера. Кроме того, мы принимаем следующие операторы:  $P\{expr\}$  оценивает вероятность внутреннего выражения (expr);  $E\{expr\}$  представляет среднее значение для внутреннего выражения (expr);  $\#r$  обозначает количество токенов на месте r. На рис. 1 показана простая компонентная модель, которая обычно используется для представления рабочего состояния компонента системы. Положение “Включено” обозначает рабочее состояние, а положение “Отключено” представляет состояние отказа. Переходы к болезни и ремонту представляют собой, соответственно, отказ и техническое обслуживание компонента. Переход “Отказ” принимает среднее время до отказа (MTTF), и “Восстановление” учитывает среднее время на ремонт (MTTR).

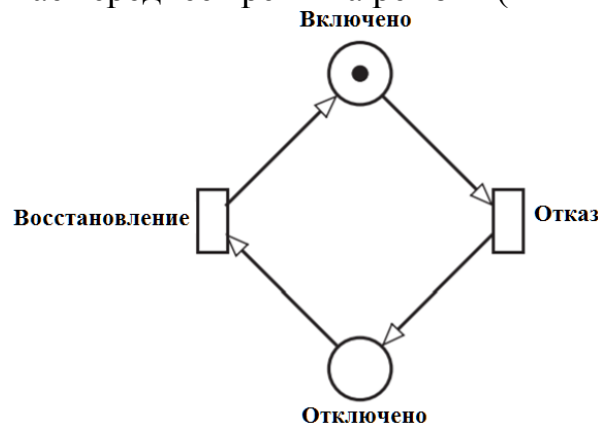


Рис. 1. Простая компонентная модель

Серверный компьютер может принять только один запрос (переход ForwardRequest), если сервер находится в рабочем состоянии (метка NodeOn). Блок контроллера использует простую компонентную модель для представления облачного контроллера, который отвечает за координацию узлов системы. Поскольку контроллер также подвержен сбоям, он также учитывается в доступности системы. Выражение защиты принимается при переходе ForwardRequest, в котором, в дополнение к доступности узла, также проверяется доступность контроллера:  $ControllerOn > 0$ . Таким образом, пропускная способность системы равна пропускной способности узла:

$$TP_{System} = TP_{node} = E\{\#ReceivingRequest\} * (1/processingDelay), \quad (1)$$

в котором processingDelay - это время, связанное с переходом по запросу.

Для обеспечения доступности системы требуется один контроллер и один узел базы данных:

$$A_{System} = P\{(\#NodeOn = 1) \& (\#ControllerOn = 1)\}. \quad (2)$$

Время простоя - еще один важный показатель, который рассчитывается с использованием недоступности системы. Время простоя (в часах) составляет:



$DT_{System} = (1 - A_{System}) * 8760$ . На рис. 2 показана система с основным узлом и двумя запасными узлами.

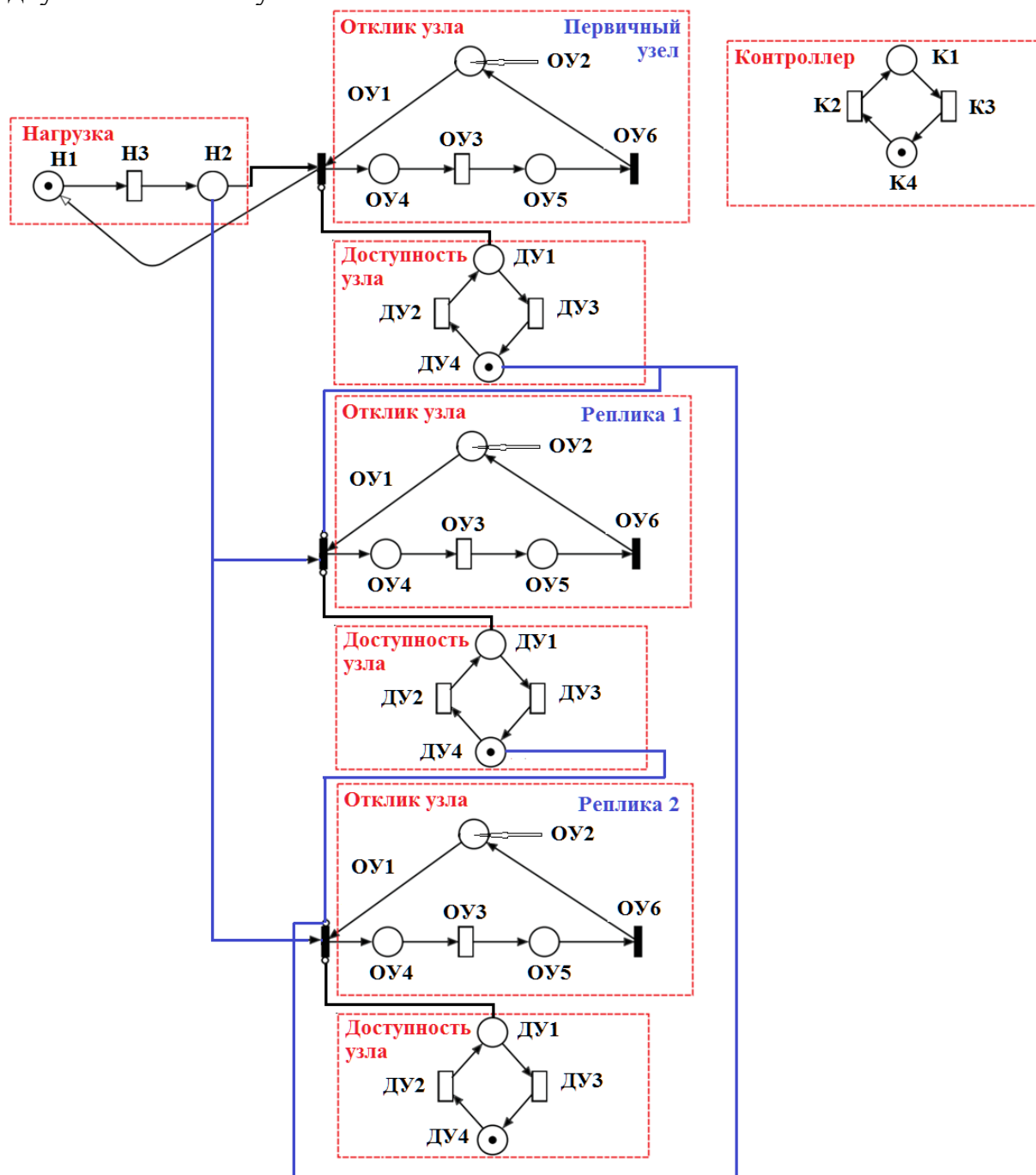


Рис. 2. Модель производительности для набора реплик

Добавляется дуга от сервера подключения места к каждому месту ForwardRequest\_Rx, и дуги-ингибиторы включаются от узла NodeOn (основного узла) к каждому переходу ForwardRequest\_Rx. Кроме того, выражения защиты проверяют доступность контроллера в каждом переходе ForwardRequest\_Rx. Дуги ингибитора деактивируют запасные узлы всякий раз, когда основной узел работает, и для модели на рис. 2 метрикой является:

$$TP_{System} = (E\{\#ReceivingRequest\} + E\{\#ReceivingRequest\_R1\} + E\{\#ReceivingRequest\_R2\}) \times (1 / processingDelay). \quad (3)$$

Доступность оценивается с учетом контроллера и, по крайней мере, одно-

го рабочего узла СУБД:

$$A_{\text{System}} = P \{((\#NodeOn = 1 \text{ ИЛИ } \#NodeOn\_R1=1 \text{ ИЛИ } \#NodeOn\_R2 = 1) \text{ И } (\#ControllerOn = 1))\}. \quad (4)$$

Для проверки предложенной модели рассмотрено частное облако на основе платформы Openstack с двумя физическими машинами: одним контроллером и одним вычислительным узлом. Вычислительный узел содержит виртуальную машину (VM) с СУБД MongoDB, представляющую подсистему хранения. Кроме того, внешняя машина используется для представления запросов клиентов. В табл.1 приведены технические характеристики.

Таблица 1

Облачные машины

Компонент	Узел контроллера	Вычислительный узел	VM	Клиент
Процессор	Intel i3-2100	Intel i7-3770	1 core	Intel i5-4200U
ОП	8GB DDR3	8GB DDR3	512MB	8GB DDR3
Жёсткий диск	350GB	250GB	10GB	SSD 250GB
ОС	Centos 7.0	Centos 7.0	Ubuntu 16.04	Centos 7.0

Рабочая нагрузка составляет 50% чтения и 50% обновления. Чтобы оценить задержку обработки и временной интервал между приходами клиентов, на виртуальной машине MongoDB было взято 30 образцов, выполнено 100000 операций и использовано переменное количество одновременных потоков (для имитации одновременных пользователей).

Таблица 2

Параметры рабочей нагрузки

Пользователи	Прибытие клиента (мс)	Задержка обработки (мс)
1	1.247	$1.16 \times 10^{-1}$
5	$2.566 \times 10^{-1}$	$9.93 \times 10^{-2}$
10	$1.717 \times 10^{-1}$	$1.03 \times 10^{-1}$
20	$1.420 \times 10^{-1}$	$1.13 \times 10^{-1}$

Таблица 3

Результаты модели для проверки пропускной способности

Прибытие клиента (мс)	Система (оп/с)	95% С.И. (оп/с)	Модель (оп/с)
1.247	801.64	[780.51; 823.72]	803.54
$2.566 \times 10^{-1}$	3897.86	[3868.92, 3939.48]	3909.64
$1.717 \times 10^{-1}$	5823.43	[5717.55, 5824.11]	5823.29
$1.420 \times 10^{-1}$	7039.27	[6788.86, 7067.13]	7038.84

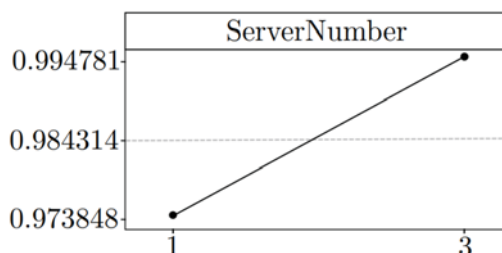


Рис. 3. График зависимости доступности от числа серверов

Набор реплик является важной функцией для облачных сред с СУБД

NoSQL, поскольку доступность значительно улучшена. Кроме того, предлагаемые модели являются важным инструментом для оказания помощи разработчикам в оценке облаков на основе NoSQL, поскольку репрезентативные системные факторы могут быть оценены без необходимости создания или прерывания работы системы для экспериментов.

**Таким образом, создана модель частных облачных вычислительных систем с подсистемами хранения данных на основе NoSQL, отличающаяся использованием обобщенных стохастических сетей Петри и обеспечивающая корректную оценку пропускной способности системы при различных рабочих нагрузках, доступности и простоях.**

Далее предполагаем, что имеется распределенная система, состоящая из множества виртуальных машин, содержащих ресурсы, и массив процессов, которые запрашивают для обслуживания доступные ресурсы, эти ресурсы ограничены, нужно отдавать приоритет одному процессу перед другим.

Шаг 1. Используя MCDM из PROMETHEE II, метод распределит ресурсы для процесса.

- для прямого критерия нормализация такова:

$$t_i^{norm} = \frac{\max_{0 \leq s \leq n}(t_s) - t_i}{\max_{0 \leq s \leq n}(t_s) - \min_{0 \leq s \leq n}(t_s)}; \quad (5)$$

- для косвенного критерия нормализация такова

$$ar_i^{norm} = \frac{ar_i - \min_{0 \leq s \leq n}(ar_s)}{\max_{0 \leq s \leq n}(ar_s) - \min_{0 \leq s \leq n}(ar_s)}. \quad (6)$$

Таблица 4

Процессы, запрашивающие ресурсы

Запрос процесса	Время выполнения (ms)	Запрошенный приоритет
$p_0$	$t_0$	$ar_0$
$p_1$	$t_1$	$ar_1$
$p_2$	$t_2$	$ar_2$
...	...	...
$p_n$	$t_n$	$ar_n$

Шаг 2: находим эволюцию, вычислив разницу по всем альтернативам, затем находим функцию предпочтения для процесса.

На следующем шаге для каждой пары  $(p_i, p_j)$  находим  $d_1$  и  $d_2$ :

$$d_1(p_i, p_j) = t_i^{norm} - t_j^{norm}; \quad (7)$$

$$d_2(p_i, p_j) = ar_i^{norm} - ar_j^{norm}, \quad \forall i, j = 0, 1, \dots, n. \quad (8)$$

Для всех критериев (прямого и косвенного) используем функцию предпочтения:

$$Q_i(p_i, p_j) = f_i(d_1(p_i, p_j)); \quad (9)$$

$$Q_2(p_i, p_j) = f_2(d_2(p_i, p_j)), \forall i, j = 0, 1, \dots, n, \quad (10)$$

где  $Q_1(p_i, p_j), Q_2(p_i, p_j) \in [0, 1]$ .

Алгоритм применяет шесть типов частных функций идентификации предпочтения. Основные:

$$1) f(d) = \begin{cases} 0, & \text{if } d \leq 0, \\ 1, & \text{if } d > 0, \end{cases} \quad f(d) = \begin{cases} 0, & \text{if } d \leq b, \\ 1, & \text{if } d > b, \end{cases} \quad f(d) = \begin{cases} 0, & \text{if } d \leq 0, \\ d/b, & \text{if } 0 < d \leq b, \\ 1, & \text{if } d > b. \end{cases}$$

Последним шагом является получение порядка процессов, которые будут обслуживаться на виртуальных машинах, путем вычисления превышения положительных и отрицательных значений:

$$\phi^+(p_i) = \frac{1}{n} \sum_{j=0}^n \pi(p_i, p_j), \quad (11)$$

$$\phi^-(p_i) = \frac{1}{n} \sum_{j=0}^n \pi(p_j, p_i), \quad (12)$$

где  $\pi(p_i, p_j) = \alpha \cdot Q_1(p_i, p_j) + (1 - \alpha) \cdot Q_2(p_i, p_j)$ ,  $\alpha$  – коэффициент важности прямого критерия. Коэффициент  $\alpha$  ( $0 \leq \alpha \leq 1$ ) получен от эксперта.

Далее найдем финальное упорядочение процессов как:

$$W_i = \phi^+(p_i) - \phi^-(p_i), \quad i = 0, 1, \dots, n. \quad (13)$$

*Алгоритм распределения ресурсов на основе двухкритериальной оценки процесса*

*Шаг 1:* Положить

$n$  – число заданий;  $m$  – число ресурсов;

$\tilde{P}$  – матрица запрашивающих ресурсов (6);

$T = (T_1, \dots, T_m)$  – вектор максимально доступных ресурсов;

$L = (L_1, \dots, L_m)$  – вектор зарезервированных ресурсов (в начальный момент запроса);

$A = (A_1, \dots, A_m)$  – вектор доступных ресурсов.

*Шаг 2:* Пусть  $q_i$  – номер  $p_i$  в Очереди ожидания;  $q = 0$  – длина очереди ожидания;

**Begin**

*Шаг 3:* Применить метод MCMD из PROMETHEE II. Получить вектор весов альтернатив  $W = (W_0, \dots, W_n)$ .

*Шаг 4:* Переупорядочить процессы в невозрастающем порядке, которые мы получаем из метода PROMETHEE II (рис. 4):

$p_{k_0}, p_{k_1}, \dots, p_{k_n}$ , где  $W_{k_0} \geq W_{k_1} \geq \dots \geq W_{k_n}$ .

*Шаг 5: For*  $h = 0$  **to**  $n$

**Begin**

Проверить процесс  $p_{k_h}$  на отсутствие дедлока с использованием метода

«Wait for Graph» или иного любого метода;  
**if** нет дедлока  
**then** распределить ресурсы для  $p_{k_h}$   
**else** отправить  $p_{k_h}$  в Очередь ожидания,  $q := q + 1$ ;  $q_{k_h} := q$ ;  
**End For**  
*Шаг 6:* **until** нет запросов в системе **then** End.

Rank	action	Phi	Phi+	Phi-
1	P0	1,0000	1,0000	0,0000
2	P2	0,2500	0,6250	0,3750
3	P3	0,0000	0,5000	0,5000
4	P1	-0,2500	0,3750	0,6250
5	P4	-1,0000	0,0000	1,0000

Рис. 4. Решение о порядке выполнения задач после применения метода MCDM PROMETHEE II

Таким образом, предложен алгоритм динамической балансировки системы, основанный на многокритериальном принятии решений при получении оптимальной матрицы альтернатив и обеспечивающий повышение производительности и QoS с одновременным определением приоритетов процессов.

**Третья глава** посвящена описанию механизмов проектирования самонастраиваемой репликации СУБД на основе форсированного обучения.

Масштабируемые облачные СУБД охватывают все большее число настраиваемых критериев, которые могут серьезно повлиять на производительность системы. Это особенно верно для реплицируемых СУБД, поскольку возможности репликации часто тесно связаны с общей конструкцией системы, и отсутствие надлежащей настройки может повлиять на надежность системы, ухудшая качество обслуживания. Кроме того, внутренние характеристики каждой рабочей нагрузки также напрямую влияют на то, как работает механизм СУБД. Предлагаемый механизм рассматривает RL, основанный на глубоком Q-обучении. В рамках подхода RL Агент отслеживает определенную среду и принимает решение о действии с помощью своей Политики. Цель агента состоит в том, чтобы найти наилучшую возможную Политику, которая максимизирует рассматриваемое вознаграждение. Значения вознаграждения вычисляются после применения Действий к Среде с использованием рассматриваемой Функции вознаграждения. Система изображена на рис. 5, 6.

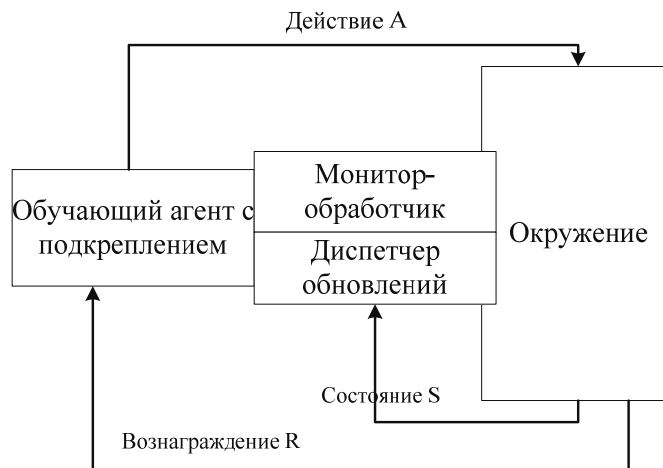


Рис. 5. Обучение с подкреплением в окружающей среде

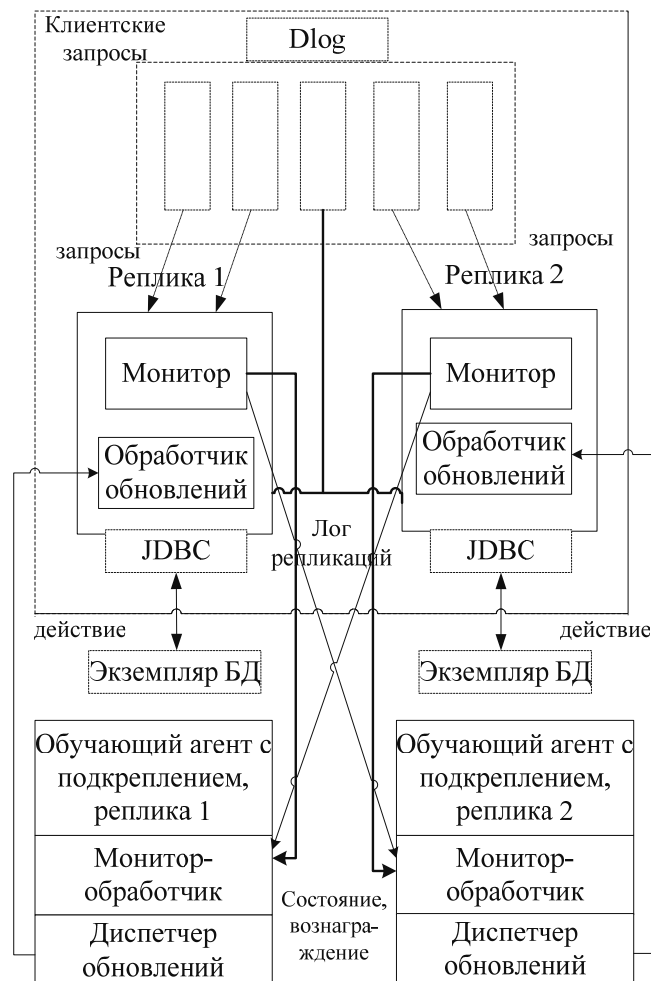


Рис. 6. Системная архитектура СУБД (обучение с подкреплением)

Оценка системы была построена с учетом эталона TPC-C, разработанного специально для оценки систем баз данных OLTP. Рабочая нагрузка определяется по 9 таблицам, управляемым набором транзакций, состоящим из пяти различных транзакций, а именно: Новый заказ, Оплата, Статус заказа, Доставка и Уровень запасов. Каждая транзакция состоит из нескольких операций чтения и обновления, где 92% составляют операции обновления, что характеризует это как большую нагрузку на запись. Соответствующая нейронная сеть приведена

на рис. 7.

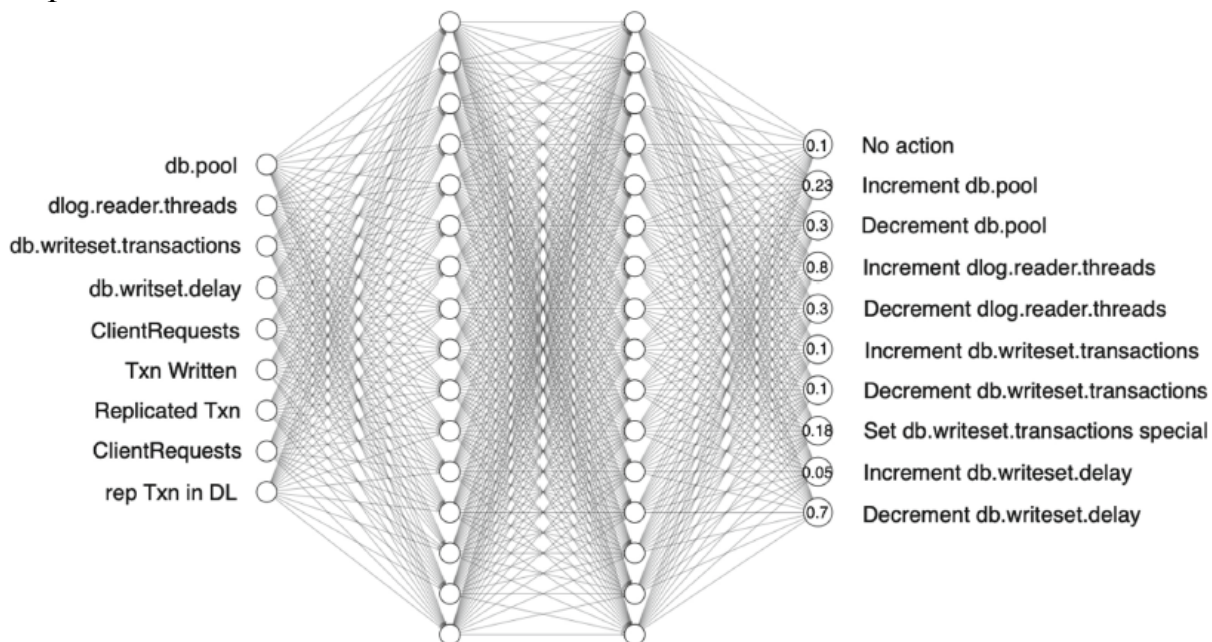


Рис. 7. Структура нейронной сети, используемой в рамках агентов обучения с подкреплением. Вычисленные значения  $q$  изображены справа в паре с соответствующим действием

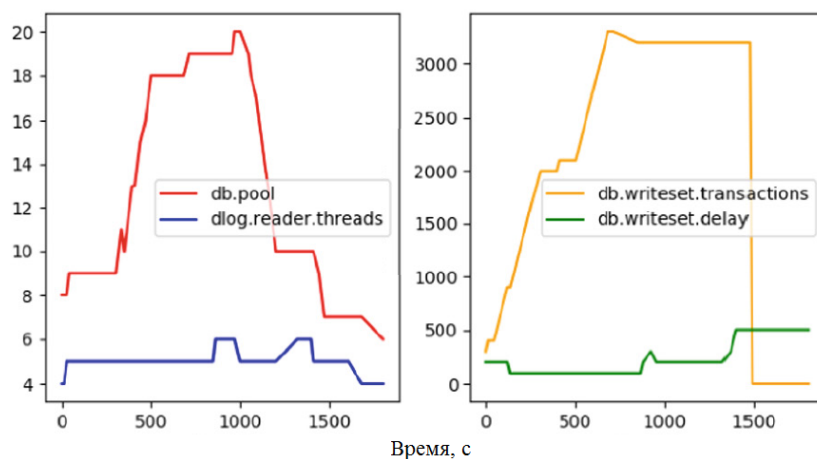
Результаты моделирования приведены на рис. 8 и в табл. 5.

Таблица 5

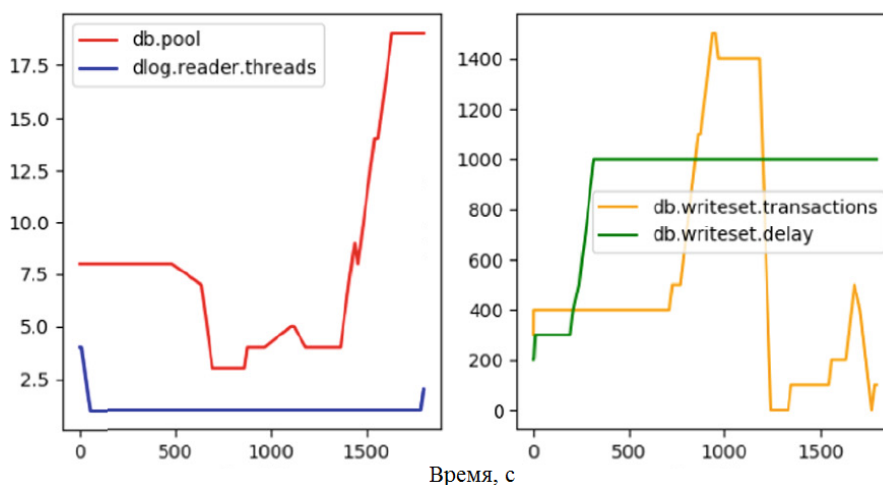
Обзор результатов

Metric	Baseline (Txn/sec)	RL10 (Txn/sec)	Прирост
ClientRequests-R1	80.80	112.95	+39.79%
Replicated Txn-R1	35.24	69.08	+96.01%
Txn Written-R1	27.57	112.95	+309.73%
ClientRequests-R2	178.20	205.47	+15.30%
Replicated Txn-R2	27.16	98.25	+261.81%
Txn Written-R2	31.86	94.26	+195.82%

Таким образом, предложен мультиагентный самонастраивающийся алгоритм репликации, основанный на обучении с подкреплением с использованием монитора метрик, диспетчера обновлений и агентов обучения с подкреплением, обеспечивающий конфигурирование гибридного промежуточного программного обеспечения репликации без перезапуска СУБД в режиме реального времени.



а) RL10 - Реплика 1



б) RL10 - Реплика 2

Рис. 8. Эволюция действий, предпринятых с переменными конфигурации во время сравнительного анализа. Цикл 10

**В главе 4** представлена архитектура распределенной защищенной много-серверной СУБД, разделяющая обработку транзакций и проектирование распределения вычислений и данных. Архитектура представлена на рис. 9.

Чтобы реализовать разрозненную организацию СУБД, мы разделяем компоненты в одной СУБД MariaDB на два различных компонента обслуживания: сервер переднего плана, обеспечивающий обработку транзакций, и серверный компонент, гарантирующий строгую защищенность и долговечность. Сервер переднего плана отправляет журналы транзакций и получает сообщения АСК от сервера внутреннего хранилища. Сообщения АСК передают надежный порядковый номер журнала (LSN), гарантирующий долговечность журналов до определенного момента. Сравнивается прототип SO-СУБД с MariaDB. Базы данных настроены на страницу данных объемом 4 КБ, пул буферов объемом 32 ГБ и минимальный уровень изоляции. На рис. 10 показана пропускная способность по оценке sysbench баз данных с различным числом клиентских подключений.



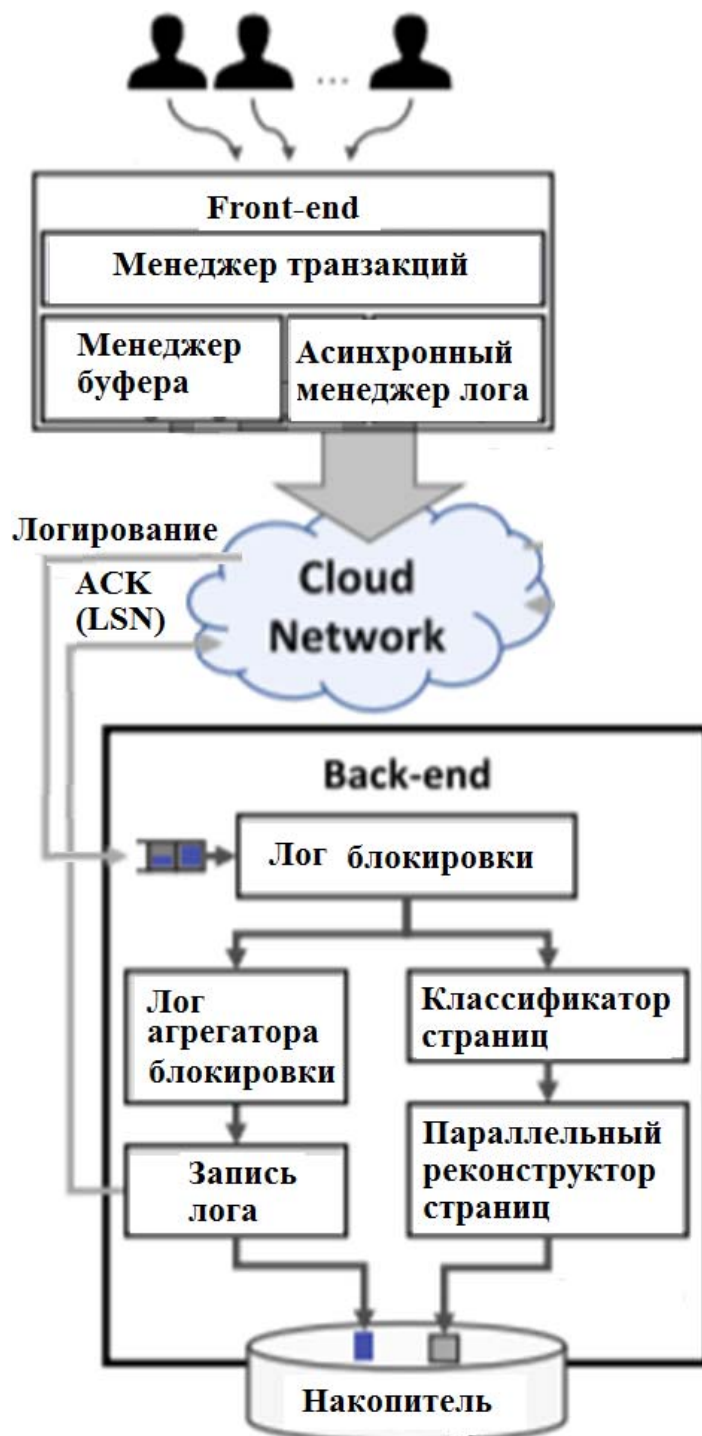


Рис. 9. Укрупненная структура прототипа SO-СУБД

Показано, что асинхронное ведение журнала уменьшает среднюю задержку фиксации на транзакцию.

Таким образом, разработана сервис-ориентированная архитектура несвязанной СУБД, отличающаяся модифицированной подсистемой ведения журнала транзакций и механизмом его очистки с нулевой памятью, обеспечивающая снижение служебного сетевого трафика и повышение безопасности обработки при проектировании разделения вычислений и данных.

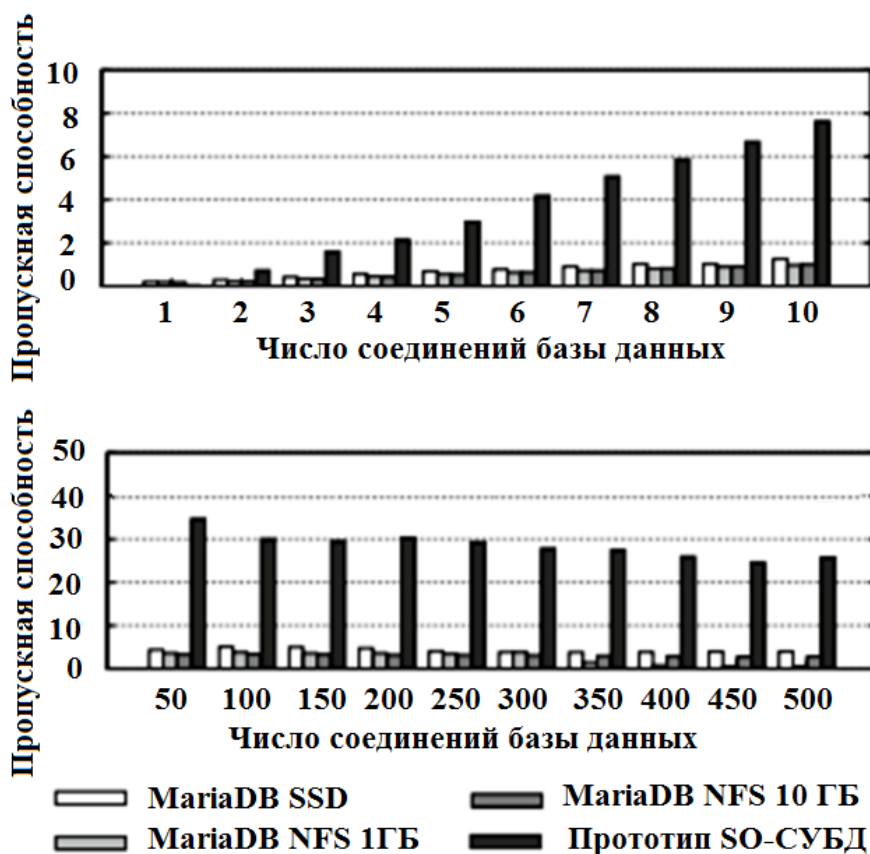


Рис. 10. Сравнение пропускной способности обработки транзакций по сравнению с эталоном sysbench

В работе рассмотрена **архитектура человеко-машинной системы проектирования базовой структуры БД**, отличающаяся интерактивным характером выбора оптимальных характеристик межсистемного взаимодействия и обеспечивающая автоматизацию динамического включения новых подмоделей в структуру БД.

Программа использует модуль StrDef, в котором определена структура для хранения информации в дереве главной формы программы. Модульная структура представлена на рис. 11, структура хранения – на рис. 12.

Основные результаты внедрены в виде программных компонент систем управления транзакциями в АО НПП «РЕЛЭКС» (г. Воронеж), «Al Rawiyah for information technology» company (Багдад, Ирак), а также в учебный процесс Воронежского государственного университета в рамках дисциплин «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

Элементы программного обеспечения зарегистрированы в ФИПС.

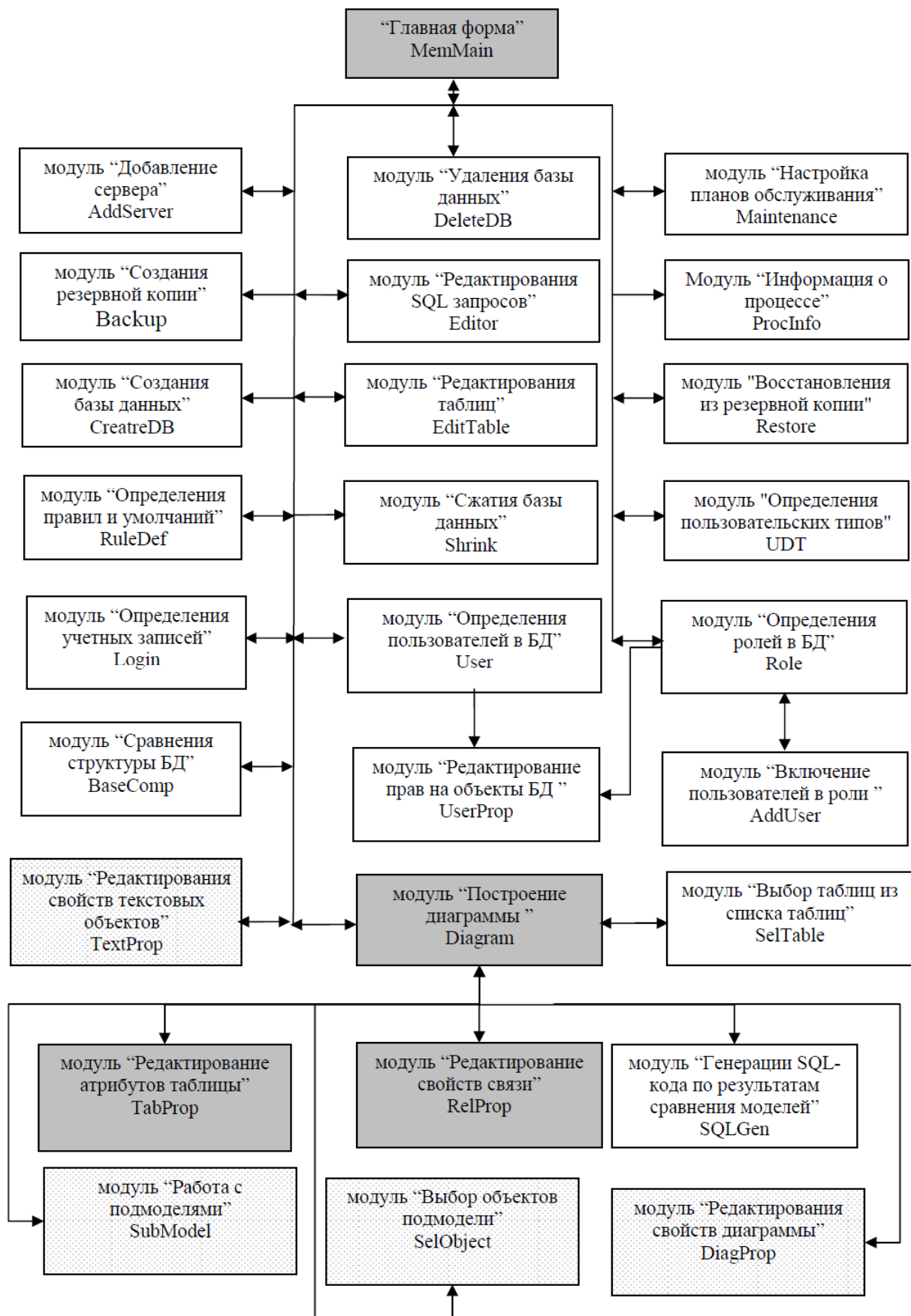


Рис. 11. Структура программного комплекса

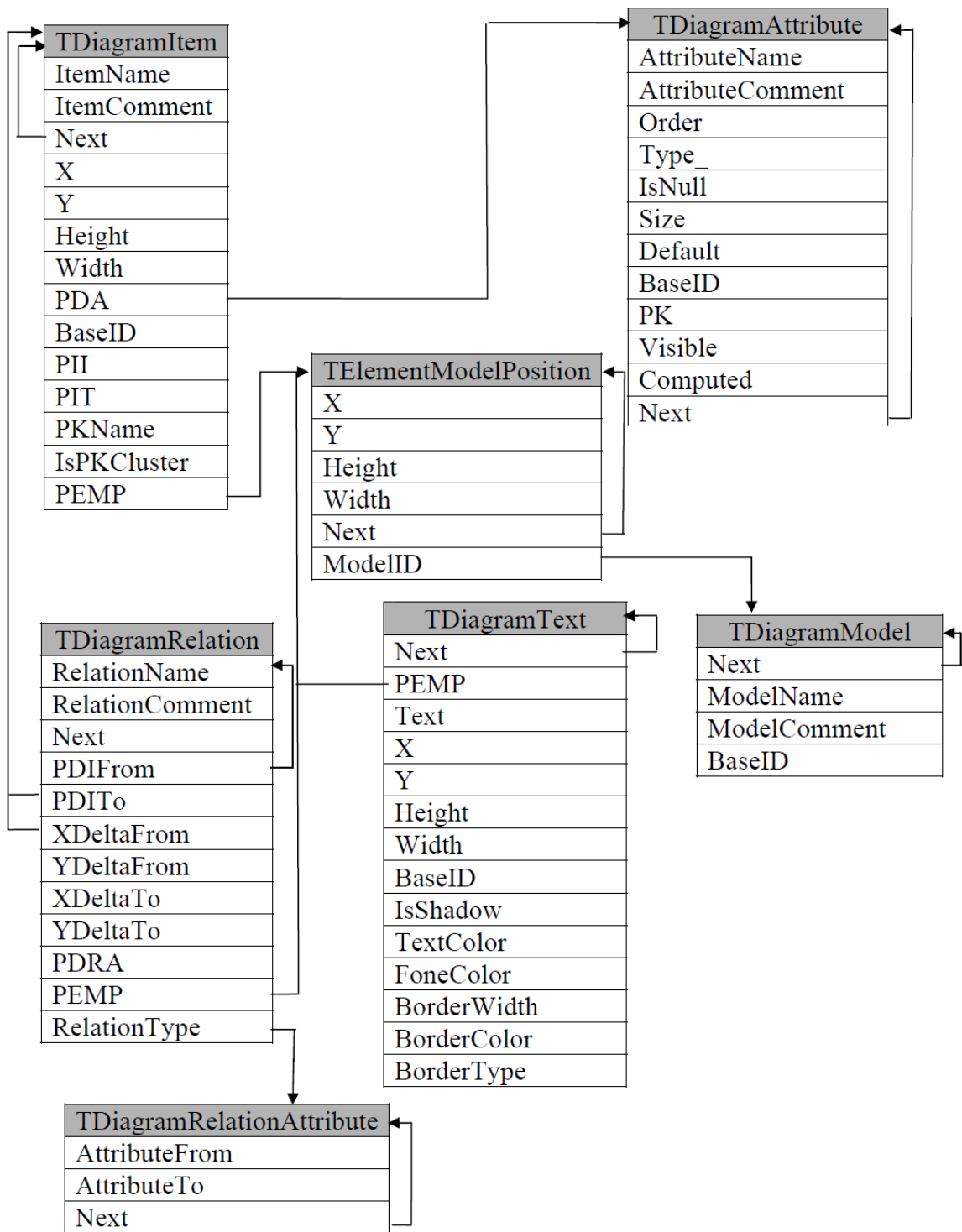


Рис. 12. Структуры хранения

### Заключение

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Создана модель частных облачных вычислительных систем с подсистемами хранения данных на основе NoSQL на обобщенных стохастических сетях Петри, обеспечивающая корректную оценку пропускной способности системы при различных рабочих нагрузках, доступности и простоях.
2. Разработан алгоритм динамической балансировки системы, обеспечивающий повышение производительности и QoS с одновременным определением

ем приоритетов процессов.

3. Разработан мультиагентный самонастраивающийся алгоритм репликации, обеспечивающий конфигурирование гибридного промежуточного программного обеспечения репликации без перезапуска СУБД в режиме реального времени.

4. Создана сервис-ориентированная архитектура СУБД, обеспечивающая снижение служебного сетевого трафика и повышение безопасности обработки при проектировании разделения вычислений и данных.

5. Представлена архитектура человеко-машинной системы проектирования базовой структуры БД, обеспечивающая автоматизацию динамического включения новых подмоделей в структуру БД.

6. Элементы программного обеспечения зарегистрированы в ФИПС.

#### **Рекомендации и перспективы дальнейшей разработки темы**

1. Результаты диссертационного исследования рекомендуются к применению в больших информационных системах, основанных на облачной обработке данных, для повышения скорости и безопасности обработки данных.

2. Дальнейшая разработка темы будет направлена на практическую реализацию теоретических и алгоритмических результатов, интеграцию в наиболее распространенные системы облачных вычислений и облачные хранилища. Безотносительно облачных терминов, развитие результатов будет направлено на улучшение безопасности репликаций, повышение эффективности алгоритмов динамической балансировки системы.

#### **Основные результаты диссертации опубликованы в следующих работах:**

##### **Публикации в изданиях, рекомендованных ВАК РФ**

1. Бондаренко Ю.В. Разработка алгоритма распределения ресурсов в распределенных системах на основе двухкритериальной оценки процессов / Ю.В. Бондаренко, А.И. Азиз // Моделирование, оптимизация и информационные технологии. – 2021. – Т. 9. – № 3. URL: <https://moitvvt.ru/ru/journal/pdf?id=1025> (DOI: 10.26102/2310-6018/2021.34.3.016)

2. Азиз А.И. Автоконфигурируемая репликация СУБД со специальным обучением для повышения безопасности / А.И. Азиз // Системы управления и информационные технологии.– 2021. – Т. 86. – №4. – С. 50-55.

3. Азиз А.И. Архитектура многосерверной СУБД с отдельной обработкой транзакций и распределением вычислений / А.И. Азиз // Системы управления и информационные технологии. – 2022.– Т. 89. – №3. – С. 66-71.

##### **Публикации в изданиях, индексируемых в WoS и Scopus**

4. Azeez A.E. Algorithm for CPU resource allocator case study and comparing between ordinary and ML Algorithms / A.E. Azeez, Yu.V. Bondarenko // IOP Conf. Ser.: Mater. Sci. Eng., 2020, 928(3). – P. 032064. (DOI 10.1088/1757-899X/928/3/032064)

5. Bondarenko Yu.V. Algorithm and model for improve the avoiding of deadlock with increasing efficiency of resource allocation in cloud environment/ Yu.V. Bondarenko, A.E. Azeez // Journal of Physics: Conference Series. – 2021. – 1902(1). – P. 012054. (DOI 10.1088/1742-6596/1902/1/012054)

## **Свидетельство о государственной регистрации программы для ЭВМ**

6. Свидетельство о регистрации программы для ЭВМ 2020664335. Согласованное распределение ресурсов в интегрированных структурах / Ю.В. Бондаренко, А.И. Азиз, О.В. Бондаренко. – № 2020663633; заявл. 03.11.2020; опубл. 11.11.2020.

### **Статьи и материалы конференций**

7. Бондаренко Ю.В. Проблемы оптимизации и самозащиты репликаций при размещении заданий в облачных вычислениях / Ю.В. Бондаренко, А.И. Азиз // Достижения науки и технологий-ДНиТ-2021: тр. Всерос. науч. конф. – Красноярск, 2021. URL: [http://ru-conf.domnit.ru/media/filer\\_public/75/1f/751ff3d0-e0e8-46ad-9efd-ba7c39fdb42d/3005-dnit-2021.pdf](http://ru-conf.domnit.ru/media/filer_public/75/1f/751ff3d0-e0e8-46ad-9efd-ba7c39fdb42d/3005-dnit-2021.pdf).

8. Азиз А.И. Механизм обучения автоконфигурируемой репликации СУБД и результаты вычислительного эксперимента / А.И. Азиз // Информационные технологии моделирования и управления. – 2021.– №4(126). – С. 293-307.

9. Azeez A.E. Algorithm and model for improve the avoiding of deadlock with increasing efficiency of resource allocation in cloud environment by using process time execution / A.E. Azeez, Yu.V. Bondarenko // Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч. конф. – Воронеж: ВГУ, 2021. – С. 493-498.

10. Azeez A.E. Research on designing and optimizing the reliability of transaction processing in a service-oriented DBMS / A.E. Azeez // Modern informatization problems in economics and safety (MIP-2022'ES): Proceedings of the XXVII-th International Open Science Conference (Yelm, WA, USA, January 2022). – Yelm, WA, USA: Science Book Publishing House, 2022. – P. 20-40.

11. Bondarenko Ju.V. Algorithm for improving the resource allocation depending on the jobs request criteria / Ju.V. Bondarenko, A.I. Azeez // Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч. конф. - Воронеж: ВГУ, 2022. – С. 1697-1700.

---

Подписано в печать 04.08.2023.

Формат 60x84/16. Бумага для множительных аппаратов.

Усл. печ. л. 1,0. Тираж 80 экз. Заказ №

Отдел оперативной полиграфии издательства ВГТУ

394006 Воронеж, ул. 20-летия Октября, 84