

Научная статья

УДК 004.942; DOI: 10.61260/2218-13X-2024-1-123-138

ОСОБЕННОСТИ ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ ПРИ УПРАВЛЕНИИ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ НА ОСНОВЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

✉ **Доронина Юлия Валентиновна;**

Мустафаева Мерьем Ибраимовна.

Севастопольский государственный университет, г. Севастополь, Россия

✉ YVDoronina@sevsu.ru

Аннотация. Рассмотрены принципы построения имитационной модели для исследования влияния вида транзакций к системе управления базой данных на прием и усвоение данных от беспилотных летательных объектов. В предложенном исследовании рассматривается взаимодействие беспилотных летательных объектов с приемником данных при реализации процесса мониторинга. Моделируется время взаимодействия беспилотного аппарата с диспетчерским центром на основе различных видов транзакций к базе данных с учетом дифференцирования нагрузки сервера системы управления базами данных и их воздействие на время обратной связи с беспилотным летательным аппаратом, что позволит оценить риски управления в оперативном режиме.

Ключевые слова: беспилотный летательный аппарат, имитационное моделирование, система управления базами данных, конфигурация производительности сервера, транзакция, управление

Для цитирования: Доронина Ю.В., Мустафаева М.И. Особенности обработки больших объемов данных при управлении беспилотными летательными аппаратами на основе имитационного моделирования // Науч.-аналит. журн. «Вестник С.-Петерб. ун-та ГПС МЧС России». 2024. № 1. С. 123–138. DOI: 10.61260/2218-13X-2024-1-123-138.

Scientific article

FEATURES OF PROPROCESSING LARGE AMOUNTS OF DATA WHEN CONTROLLING UNMANNED AERIAL VEHICLES BASED ON SIMULATION MODELING

✉ **Doronina Yulia V.;**

Mustafaeva Merem I.

Sevastopol state university, Sevastopol, Russia

✉ YVDoronina@sevsu.ru

Abstract. The paper discusses the principles of constructing a simulation model to study the influence of the type of transactions to a database management system on the reception and assimilation of data from unmanned aerial objects. The proposed study examines the interaction of unmanned aerial objects with a data receiver when implementing the monitoring process. The interaction time of the unmanned aerial vehicle with the dispatch center is modeled based on various types of transactions to the database, taking into account the differentiation of the database management system server load and their impact on the feedback time from the unmanned aerial vehicle, which will allow assessing the risks of control in an online mode.

Keywords: unmanned aerial vehicle, simulation modeling, database management system, server performance configuration, transaction, management

For citation: Doronina Yu.V., Mustafaeva M.I. Features of processing large amounts of data when controlling unmanned aerial vehicles based on simulation modeling // Scientific and analytical journal «Vestnik Saint-Petersburg university of State fire service of EMERCOM of Russia». 2024. № 1. P. 123–138. DOI: 10.61260/2218-13X-2024-1-123-138.

Введение

Современные беспилотные летательные аппараты (БПЛА) применяются для решения задач различных типов, одним из которых является сбор данных мониторинга объектов или их параметров в форме изображений, видео фиксации объектов или явлений, геолокационных данных этих объектов и т.п. Эти данные имеют огромный потенциал для применения в разведке, наблюдения за окружающей средой, поставки и транспортировки товаров, медицинских исследований. Высокоточная обработка и анализ высоконагруженных данных представляют серьезные вызовы, особенно когда речь идет о реальном времени и высокой степени точности [1].

Сбор мониторинговых данных может быть осуществлен как дискретно, так и непрерывно – потоком. В первом случае, на входе приемника данных собираются: цифровая информация, изображения, относящиеся к конкретному моменту времени существования конкретного объекта, описывающие одну или несколько характеристик этого объекта или явления. Например, изображение сельскохозяйственной площади с привязкой к координатам и параметрами интенсивности цветовой насыщенности, что может быть в дальнейшем использовано при принятии решения о качестве процесса роста сельскохозяйственной агрокультуры [2]. Во втором случае, данные от БПЛА могут быть потоковыми, в виде видеоряда, что наиболее часто применяется, например, для оценки динамики существования объекта или явления, при слежении, мониторинге и оперативных решениях, причем здесь важна идентификация объекта мониторинга, который может быть статическим или динамическим. В случае динамического (подвижного) объекта мониторинга, собираемые данные зачастую также динамические, потоковые, возможно, разнообразные. Например, при мониторинге передвижения льда в акватории, видеоряд может быть однообразным, а при слежении за движущимся хищником на пастбище, скорость принятия решений при формировании отпугивающих сигналов, исходя из меняющейся ситуации, должна быть максимальной.

Что касается экстренных ситуаций, БПЛА могут предоставить пользу предотвращения пожаров. БПЛА легко выполняют мониторинг больших территорий, обеспечивая оперативное обнаружение начальных признаков пожара. Автоматизированный мониторинг БПЛА выявляет пожар на ранних стадиях, позволяя быстро реагировать и принимать решения, предотвращать распространение.

Данные, которые БПЛА собирает, сохраняют в базу данных (БД) для анализа. Это дает возможность определить масштаб пожара, его направление, интенсивность, что важно в экстренных ситуациях. БПЛА используются при эвакуациях, отслеживают эвакуацию людей и животных, обеспечивая безопасность жизни. Важные данные могут быть переданы службам экстренной помощи для планирования и координирования действий.

БПЛА с технологией искусственного интеллекта принимает решения на основе данных, строит оптимальные маршруты тушения огня или определения критических участков тушения огня при срочном вмешательстве.

БПЛА оперативен в действии при борьбе с пожарами, а также в минимизации рисков для людей, работающих в условиях чрезвычайной ситуации.

На рис. 1 приведена классификационная схема, отражающая типы и требования к взаимодействию БПЛА-приемник.



Рис. 1. Типы и требования к взаимодействию БПЛА-приемник

В приведенных на рис. 1 требованиях к взаимодействию БПЛА-приемник операции БПЛА требуют обработки данных и систематическое сохранение полученных сведений в БД посредством взаимодействия с оператором диспетчерского центра. Кроме того, при таком взаимодействии требуется интенсивный перерасчет координат для оптимизации движения БПЛА и взаимодействие нескольких подсистем: диспетчера, системы принятия решений, подсистем технического зрения, стратегий выполнения конкретных операций мониторинга, подсистемы навигации и др. При работе БПЛА генерируется большое количество данных, включая координаты полета, фотографии и видео, которые передаются в хранилище данных. Обработка этих данных позволяет сформировать вторичные данные, которые применимы для принятия решений по ряду процессов мониторинга.

Таким образом, не только эффективное получение, но и качественное усвоение больших объемов данных от БПЛА являются важными задачами при мониторинге в различных областях. Под усвоением данных в статье понимается процесс взаимодействия БПЛА-приемник, исходя из чего решается задача анализа и выявления особенностей обработки больших объемов данных от БПЛА при мониторинге в зависимости от использования различных конфигураций транзакций, содержащих разнообразные запросы к системе управления базами данных (СУБД) на основе имитационного моделирования.

Целью статьи является построение имитационной модели для исследования влияния вида транзакций к СУБД на прием и усвоение данных от БПЛА с целью поддержки принятия решений при управлении параметрами передачи данных от БПЛА при мониторинге объектов или явлений различной природы.

Для достижения данной цели решаются следующие основные задачи:

1. Анализ роли СУБД в обработке и хранении данных, а также разработка оптимальных подходов к интеграции СУБД в системы обработки данных для БПЛА.
2. Экспериментальное исследование влияния вида транзакций на производительность серверов и их воздействие на время обработки и усвоения данных об объекте мониторинга.
3. Практические результаты и рекомендации для оптимального использования ресурсов серверов СУБД при усвоении данных от БПЛА и принятие решений по управлению параметрами БПЛА.

Обзор существующих решений и методика исследований

В предложенном исследовании рассматривается взаимодействие БПЛА с приемником данных при реализации процесса мониторинга. Сам объект исследования и вид БПЛА не учитывается. Внимание авторов уделяется исследованию влияния различных конфигураций транзакций на производительность серверов СУБД и их воздействие на время обработки и анализа данных от БПЛА, что и понимается под термином «усвоение данных».

В статьях [3, 4] проведены исследования параллельных систем БД. Авторы рассматривают применение графических ускорителей и многоядерных сопроцессоров в параллельных системах БД с использованием эмулятора; исследование позволяет оценить эффективность этой технологии. В источниках [4–6] представлены математические модели обработки запросов поиска, вставки и удаления в динамических БД. Общая модель, применяемая в этих работах, позволяет оценить сложность решения задач и разрабатывать распараллеливаемые алгоритмы в задачах данного класса, что в целом может быть элементом технологии усвоения данных мониторинга, но не учитывает особенности связки БПЛА-приемник.

В работе [7] предложена идея использования имитационного моделирования для определения оптимального размера битового индекса в реляционных БД. Результаты исследования позволяют архитекторам БД оптимизировать индексы, учитывая особенности бизнес-процессов, а в источниках [8–10] охватываются современные методы геоинформационного анализа, использующие базовые и дополнительные модули современных геоинформационных систем. В исследованиях представлена обобщенная схема проведения пространственного анализа данных, приведены практические примеры использования геоинформационного анализа для мониторинга и моделирования различных пространственных структур.

В источниках [11–17] рассматривается проблематика применения небольших гражданских БПЛА для обнаружения стихийных бедствий. Предложена новая платформа приложений для БПЛА с облачной поддержкой и прототип реализации. Платформа объединяет сбор видео, планирование данных, выгрузку и обработку данных, а также измерение состояния сети, создавая эффективную и масштабируемую систему. Прототип платформы состоит из клиентского набора компонентов, размещенных на БПЛА, который выборочно выгружает собранные данные на облачный сервер. Сервер обеспечивает обработку данных в режиме реального времени и обратную связь с центром управления инцидентами и клиентским устройством/оператором.

В статьях [18–23] определена роль в управлении кризисами и усилиях по оказанию помощи в случае стихийных бедствий. Новое поколение БПЛА может помочь улучшить ситуационную осведомленность и оценку информации. Среди преимуществ БПЛА – освобождение спасательных команд от трудоемких задач по сбору данных и помощь исследовательским операциям с более глубоким и точным руководством, благодаря расширенным сенсорным возможностям. В этих научных сведениях обсуждаются потенциалы БПЛА в оказании помощи в различных сценариях оказания гуманитарной помощи, а также возможные проблемы в таких ситуациях.

Таким образом, существующие исследования в области производительности БД при реализации процесса обработки данных дистанционного мониторинга в целом имеют серьезное основание, обеспечивая новые методы и инструменты для оптимизации и повышения производительности систем обработки данных. Однако для принятия комплексных решений при усвоении мониторинговых данных объектов различной природы больших объемов от БПЛА (от управления параметрами сбора данных до изменения типа запроса в БД в рамках конкретной конфигурации СУБД) требуется дополнительная проработка и предложенное исследование на основе имитационного моделирования [1, 24] является актуальным.

Типовой алгоритм обработки разных типов заявок от БПЛА и влияние на производительность СУБД

Исследование разнообразных конфигураций потока заявок при анализе производительности системы на основе имитационного моделирования позволяет оптимизировать ресурсы, предсказывать производительность системы и принимать информированные решения при планировании, настройке и управлении системами [25–27].

В исследовании предложено классифицировать сервер СУБД в трех конфигурациях: производительный, стандартный, слабый. Это позволит варьировать (перенаправлять) возможные серии транзакций для увеличения производительности усвоения данных в случаях необходимости оперативной обработки или интенсивной динамики объекта или самого БПЛА.

Рассмотрим предложенный алгоритм обработки разных типов заявок от БПЛА и влияние их на производительность сервера СУБД.

1. БПЛА генерирует поток событий:

- поиск новых объектов;
- полет до границы загруженного региона на карте;
- реакция на ситуационные обстоятельства, например, сильный ветер, изменения в параметрах движения (представляющие собой линию от самолета до выбора объекта).

2. Блок управления БПЛА реагирует на событие созданием запроса к базе и выполнением операций:

- скачивание новых фрагментов карты. Установим признак 3 (производительный сервер);
- уточнение параметров региона. Установим признак 3 (производительный сервер);
- запись данных о новых объектах. Установим признак 3 (производительный сервер);
- запрос на получение метеорологических данных. Установим признак 3 (производительный сервер);
- запрос на выполнение разведки или наблюдения. Установим признак 2 (стандартный сервер);
- запрос на изменение маршрута полета. Установим признак 2 (стандартный сервер);
- запрос на обновление программного обеспечения. Установим признак 1 (слабый сервер).

3. В ответ на поток событий блок управления выбирает один или несколько типов запросов, которые будут отправлены к БД, или может решить не выполнять никаких действий, проигнорировав событие. Это решение зависит от конкретного типа запроса к базе, который должен быть выполнен в ответ на внешние события.

4. При передаче запросов возможна потеря данных из-за ненадежной сети. Для решения этой проблемы, в модель добавлены условные блоки и пути возврата «1» и «2». Если происходит ошибка при передаче запроса или ответа, запрос считается «потерянным», но это событие, то есть «запрос был потерян», возвращается в блок управления. В ответ на это событие блок управления может сгенерировать новый запрос или принять решение проигнорировать его.

На основе предложенного алгоритма разработана имитационная модель в среде AnyLogic, для уточнения настроек которой далее приводится описание имитируемого процесса с параметрами.

Имитационная модель в среде AnyLogic для анализа влияния категоризации транзакций в форме запроса SQL на производительность БД

Имитируемый процесс представляет собой модель взаимодействия источника (в форме запросов от БПЛА через диспетчерский центр к СУБД, далее с целью сокращения промежуточных записей – просто от БПЛА к СУБД, преобразованных в SQL) и приемника

(БД, сервера СУБД), на основе, в том числе блоков обработки задержек, которые используются в качестве установщиков параметров имитационного цикла.

Последовательность процесса работы модели следующая.

1. Источник генерирует входной поток запросов с различными типами обработки данных (могут быть: простая выборка (в форме SELECT по простому условию поиска), s_0 ; сложный ресурсозатратный запрос, включающий операцию соединения отношений БД в виде запроса типа JOIN, s_{Join} ; запрос с большим объемом ответа, например, blob-поля с данными карты) и каналами обработки, s_{blob} . Каждый тип запросов характеризуется своим набором параметров:

- а) трудоемкостью обработки запроса в СУБД, α ;
- б) объемом передаваемых данных в запросе (от БПЛА к СУБД), v_b ;
- в) объемом данных в ответе (от СУБД к БПЛА), v_i .

Сложный запрос с join характеризуется повышенной трудоемкостью обработки, а запрос с blob – повышенным временем передачи ответа по сети. Параметры разных типов генерируются с разной периодичностью: например, запрос за участками карты – периодически с установленным периодом, JOIN – через случайные промежутки времени с показательным распределением. Таким образом, моделируются периодические и случайные процессы в работе БПЛА.

Поток данных Z_t от источника (с учетом времени формирования запроса t и до оценки производительности в следующий момент времени $t+1$) от заявок имеет вид:

$$Z_t := (s_0 \vee s_{Join} \vee s_{blob})_t | (\alpha, v_b, v_i)_{t+1} >. \quad (1)$$

Выражение (1) распространяется на все типы серверов, в общем случае имеющие конфигурации: производительный (w_2), стандартный (w_1), слабый (w_0). Тогда справедливо семейство представлений Z_t :

$$\begin{aligned} Z_t^{w_0} &:= (s_0 \vee s_{Join} \vee s_{blob})_t | (\alpha^{w_0}, v_b, v_i, \xi^{w_0})_{t+k_0} >, \\ Z_t^{w_1} &:= (s_0 \vee s_{Join} \vee s_{blob})_t | (\alpha^{w_1}, v_b, v_i, \xi^{w_1})_{t+k_1} >, \\ Z_t^{w_2} &:= (s_0 \vee s_{Join} \vee s_{blob})_t | (\alpha^{w_2}, v_b, v_i, \xi^{w_2})_{t+k_2} >, \end{aligned}$$

где $\alpha^{w_0}, \alpha^{w_1}, \alpha^{w_2}$ – трудоемкости обработки различных запросов s_0, s_{Join}, s_{blob} на определенных типах конфигурации серверов; $\xi^{w_0}, \xi^{w_1}, \xi^{w_2}$ – управляемые параметры, влияющие на производительность каждой конфигурации серверов СУБД; k_0, k_1, k_2 – интервалы времени, отражающие производительность каждой конфигурации серверов СУБД.

Требуется построить имитационную модель для исследования влияния вида транзакций к СУБД на прием и усвоение данных от БПЛА с целью поддержки принятия решений при управлении параметрами передачи данных от БПЛА при мониторинге объектов или явлений различной природы.

На рис. 2 приведены некоторые программные параметры имитационной модели, реализованной по предложенному алгоритму в среде AnyLogic.

2. Блок очереди сети обрабатывает поступающие данные по параметрам:

- а) признак входного параметра = генерация входных запросов;
- б) признак очереди = значение;
- в) признак выходного параметра = значение;
- г) значение метод size – возвращает количество элементов;
- д) количество интервалов – 10.

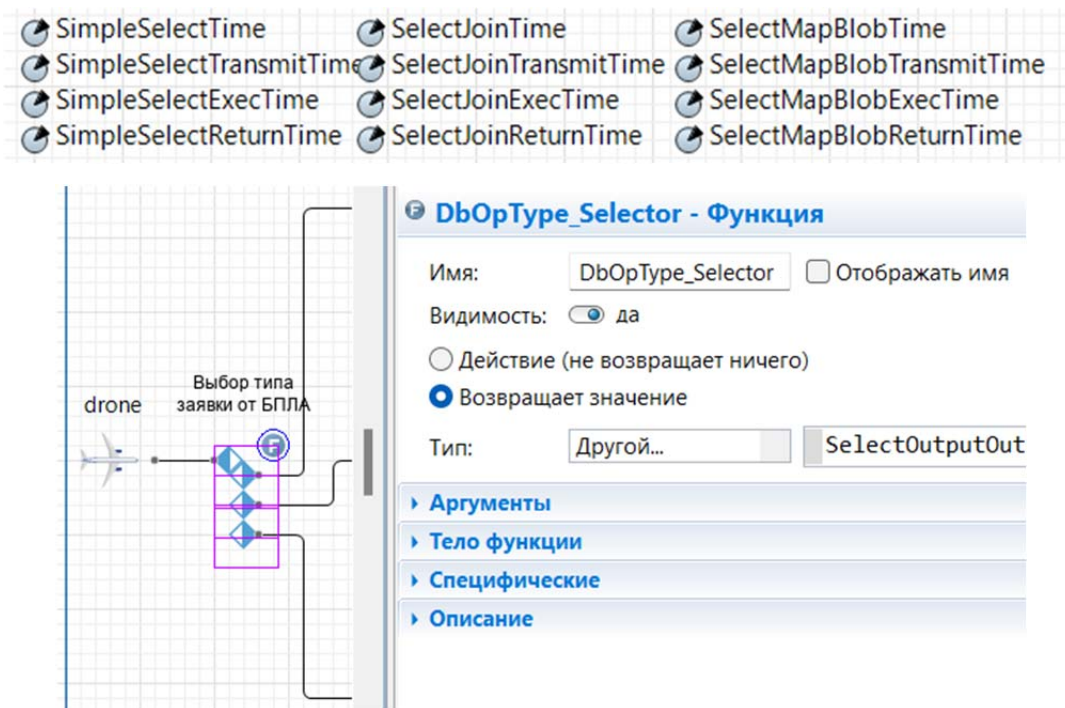


Рис. 2. Параметры модели при имитации взаимодействия БПЛА и типов запросов при генерировании источника входных параметров

3. Выходные параметры попадают в блок задержки:

- а) задержка по времени = текущее время задержки;
- б) время задержки определяется как объем данных в запросе деленное на пропускную способность канала передачи данных;
- в) разные величины пропускной способности канала позволяют моделировать различные типы каналов передачи данных.

4. Блок очереди БД обрабатывает данные, поступающие с блока задержки сети:

- а) признак входного параметра = генерация входных запросов;
 - б) признак очереди = значение;
 - в) признак выходного параметра = значение;
- г) значение метода size – возвращает количество элементов;
д) количество интервалов – 10.

5. Выходные параметры поступают в блок таймера задержки БД:

- а) задержка по времени = текущее время задержки;
- б) время задержки определяется как трудоемкость заявки, деленное на производительность СУБД;
- в) разные величины производительности СУБД позволяют моделировать различные типы БД и используемого оборудования для их исполнения.

6. Блок ответа очереди сети измеряет время обработки и возвращает данные.

7. Аналогично время задержки ответа по сети определяется как объем данных в ответе, деленное на пропускную способность канала передачи данных. Блок ответа БД фиксирует затраченное время обработки и возвращает данные.

Общий вид модели имитации взаимодействия БПЛА с учетом категоризации транзакций в форме запросов SQL приведен на рис. 3.

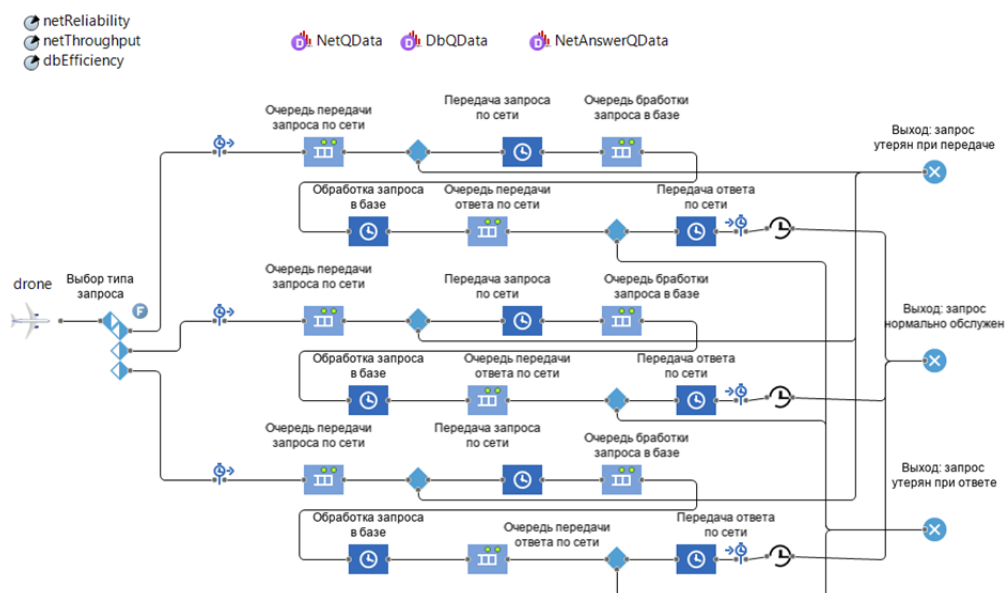


Рис. 3. Общий вид модели имитации взаимодействия БПЛА с приемником с учетом категоризации транзакций в форме запросов SQL

Для описания работы модели и результатов её выполнения введем следующие обозначения: R (Request) – запрос к БД; C (Channel) – канал; h (heavy) – «тяжелый» запрос к БД; m (map) – карта; r (random) – случайность, тогда:

– R_j – сложный запрос с join, характеризующийся повышенной трудоемкостью обработки.

– R_b – сложный запрос с blob, характеризующийся повышенным временем передачи ответа по сети.

– R_h – запрос с большим объемом ответа: сложный запрос с большим ответом и загруженным процессом обработки.

– R_m – запрос по участкам карты, который происходит дискретно с установленным периодом.

– R_r – запрос join, выполняющийся через случайные промежутки времени в общем случае с показательным распределением.

– C_{in} – источник генерирует входной поток запросов.

– C_{net} – очередь сети.

– C_{delay} – блок задержки.

Формула рассчитывается следующим образом:

$$\tau = \frac{Q}{N},$$

где τ – время задержки; Q – объем данных в запросе; N – пропускная способность канала передачи данных.

– C_{db} – блок таймера задержки базы данных. Формула рассчитывается следующим образом:

$$\lambda = \frac{\Omega}{\omega},$$

где λ – время задержки определяется; Ω – как трудоемкость заявки; ω – производительность СУБД.

– C_{out} – выходные параметры поступают в блок таймера задержки БД.

Время задержки ответа по сети определяется как объем данных в ответе, деленное на пропускную способность канала передачи данных. На рис. 4 приведен вид работающей модели имитации взаимодействия БПЛА-приемник.

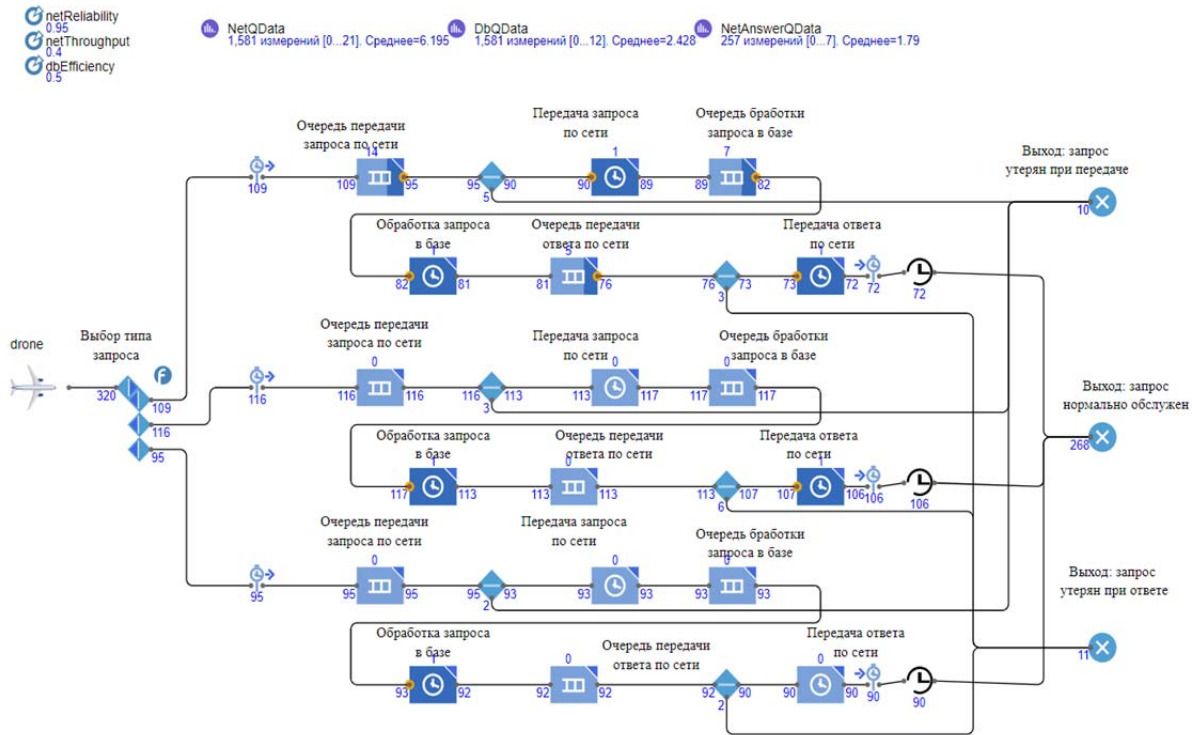


Рис. 4. Вид работающей модели имитации взаимодействия БПЛА-приемник с промежуточными результатами

Обсуждение результатов имитационного моделирования

Результаты тестирования имитационной модели (рис. 3) представлены на рис. 5, 6. Вертикальная линия на графиках отображает среднее значение соответствующего распределения.

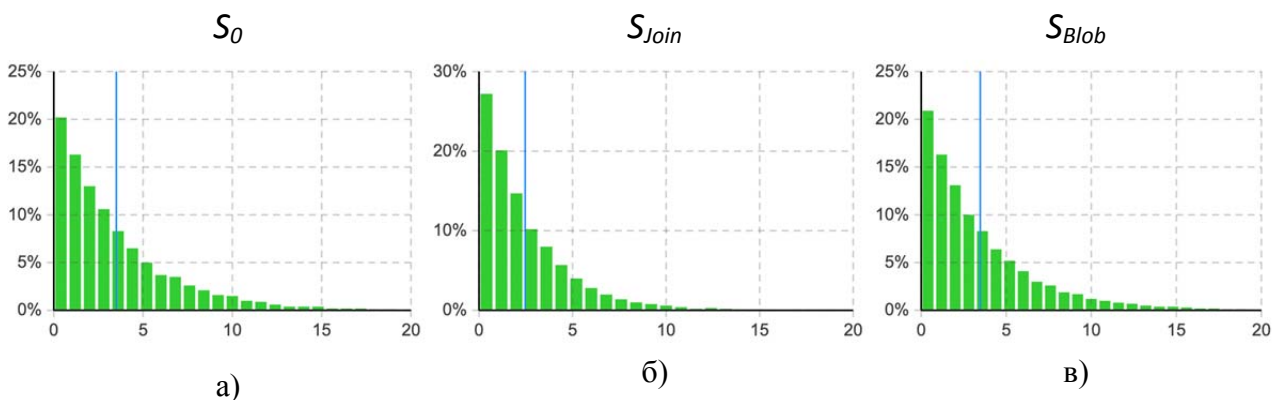


Рис. 5. Результаты тестирования имитационной модели: распределение времени между обработанными заявками по типам на выходе из модели

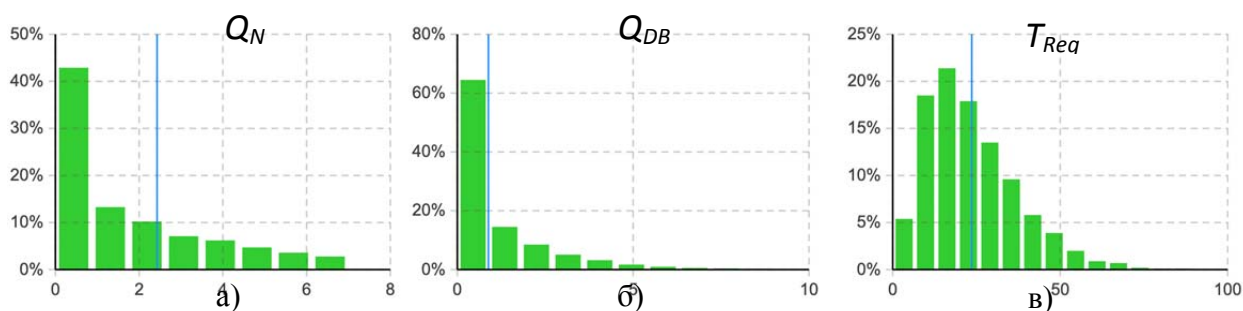


Рис. 6. Результаты тестирования имитационной модели: распределения длины очереди сети, длины очереди выполнения запроса в БД и распределение полного времени обслуживания заявок

На рис. 5 приведены графики распределения времён между заявками на выходе из модели, то есть после полного их обслуживания на всех этапах (передача по сети, выполнение запроса, возврат ответа по сети) по каждому из типов заявок. Можно установить, что большинство простых запросов S_0 обрабатывается в короткие временные промежутки (рис. 5 а), что отражает низкую трудоемкость и высокую скорость обработки таких запросов сервером. Вид распределения времен между запросами с операциями соединения (рис. 5 б) показывает, что, хотя многие запросы типа S_{Join} также выполняются достаточно быстро, существует широкий разброс временных интервалов, что указывает на более сложную природу таких запросов и возможное разнообразие временной нагрузки на СУБД. Результаты, приведенные на рис. 5 в, отражают ситуацию, когда запросы с большим объемом данных S_{Blob} требуют более продолжительной обработки, тем не менее большая часть их также обрабатывается в сравнительно короткий период, что свидетельствует о достаточном уровне оптимизации сервера для работы с объемными данными.

Из полученных данных видно, что при конфигурировании СУБД необходимо учитывать разнообразие типов запросов. Для улучшения производительности системы может потребоваться адаптация ресурсов под наиболее ресурсоемкие операции, а также использование стратегий кэширования и приоритезации запросов для эффективной обработки данных от БПЛА. Эти результаты подчеркивают важность детального анализа и внимания к конфигурации серверов при разработке систем, принимающих и обрабатывающих мониторинговые данные.

На рис. 6 представлены результаты имитационного моделирования, отражающие три ключевых аспекта системы обработки запросов: длину очереди сети (а), длину очереди выполнения запроса в БД (б) и распределение полного времени обслуживания заявок (в):

а) длина очереди сети Q_N представляет количество задач, ожидающих передачи в сети. График демонстрирует, что большая часть запросов быстро проходит через сеть без значительного ожидания, однако небольшое число запросов возникает в очереди, что указывает на потенциальные задержки в сетевой инфраструктуре;

б) длина очереди выполнения запроса в СУБД Q_{DB} отображает, как много запросов одновременно обрабатывается БД. Заметно, что большая доля запросов обслуживается почти мгновенно, тогда как остальные распределены равномерно. Наличие высоких значений размера очереди говорит о возможном затруднении в обработке запросов из-за высокой нагрузки на СУБД или меньшей производительности серверов;

в) распределение времени обслуживания заявок T_{Req} показывает, сколько времени требуется от начала до конца обработки запроса. Из графика видно, что большинство запросов обрабатывается в пределах коротких временных интервалов, но также существует значительное количество заявок, требующих гораздо больше времени, что может указывать на сложные или ресурсоемкие операции в запросах или неоптимальную конфигурацию СУБД.

В таблице и на рис. 7 приведены результаты моделирования вида запроса и канала обработки при условиях различных конфигураций серверов СУБД.

Таблица

Результаты моделирования вида запроса и канала обработки при условиях различных конфигураций серверов СУБД

№	Вид запроса	Канал обработки	Конфигурация 1. Производительный сервер			Конфигурация 2. Стандартный сервер			Конфигурация 3. Слабый сервер		
			1000 записи	10 000 записи	100 000 записи	1000 записи	10 000 записи	10 0000 записи	1000 записи	10 000 записи	100 000 записи
1	R_j	C_{in}	1,4	5,2	30,6	5	30,9	120,4	20,5	120,1	240
2	R_b	C_{in}	5,4	30,1	300,7	20,2	120,3	400,5	120,7	620,9	1800,1
3	R_h	C_{in}	5,3	20,5	120,2	10,4	60,5	600,4с	30,4	180,2с	1200,3
4	R_m	C_{ng}	10,2	60,1	600,2	30,3	180,4	1800,3	60,5	600,3	3600,2с
5	R_r	C_{ng}	2,3	10,2	60,2	10,1	60,6	600,8	30,3	180	1800,1
6	R_h	C_{ng}	30,3	180,1	1800,3	60,5	600,4	3600,3	180,2	1800,1	10 800,7
7	R_j	C_{delay}	0,5	2,5	25,3	2,5	12,5	125,3	12,5	62,5	625,8
8	R_b	C_{delay}	2,3	10,1	102,2	14,5	50,3	500,7	51,5	253,9	2448,2
9	R_h	C_{delay}	15,2	75	750,2	50,5	251,7	2507,5	150,1	750,2	7500,9
10	R_j	C_{db}	00.1	0,52	5,3	0,52	2,53	25,5	2,3	10,1	100,4
11	R_b	C_{db}	0,5	2,5	25,2	2,25	12,45	125,3	10,3	50,7	500,9
12	R_h	C_{db}	1,5	7,5	75,1	7,5	37,3	375,8	30,5	150,9	1500,2
13	R_j	C_{out}	0,1	1	10,5	1,9	10,6	100,5	10,2	100,4	1000,2
14	R_b	C_{out}	1,8	10,3	100,1	10,4	100,3	1000,8	100,3	1000,9	10000,3
15	R_h	C_{out}	3,2	30,0	300,2	30,1	300,6	3000,2	300,1	3000,5	30000,8

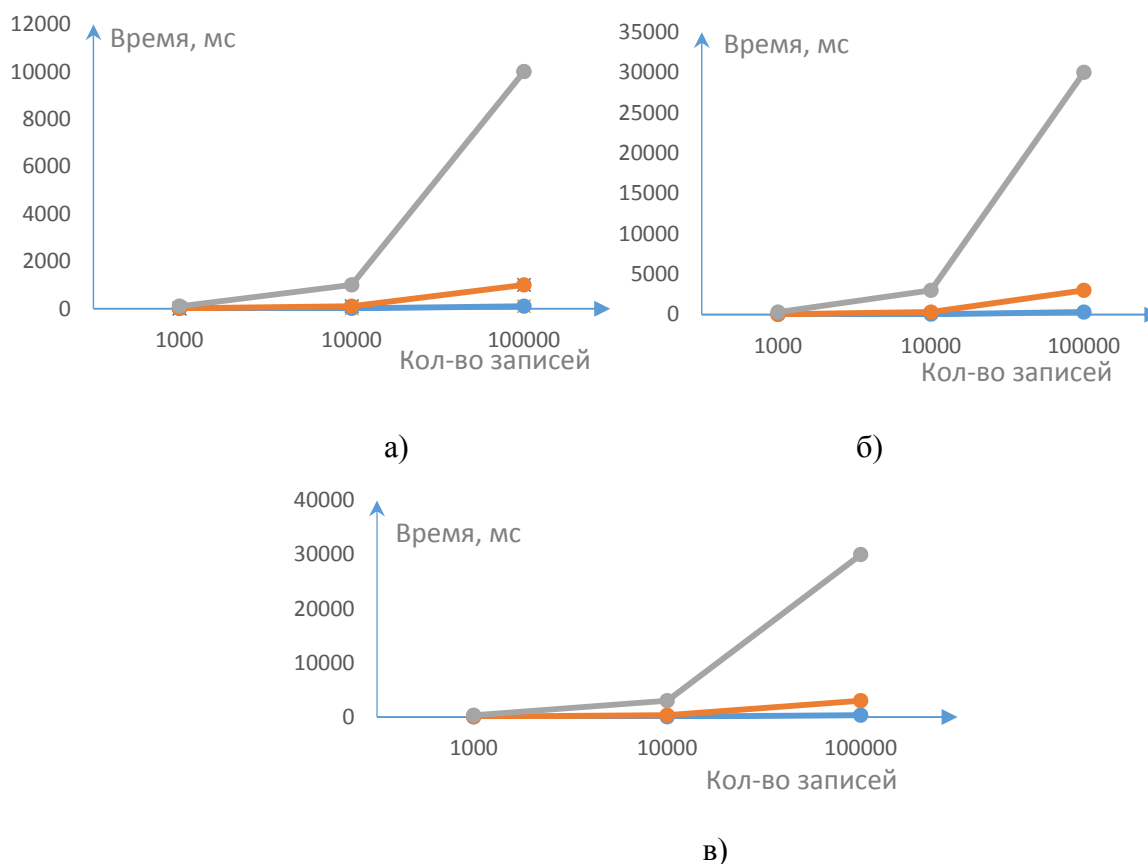


Рис. 7. Графики зависимостей времени обработки заявки от количества записей:
а) при сложном запросе с JOIN; б) при сложном запросе с blob;
в) при сложном запросе с большим ответом (блок таймера задержки БД)

На рис. 7 представлены графики зависимостей времени обработки запросов от их количества для различных типов запросов к БД: а) сложный запрос с JOIN; б) сложный запрос с blob; в) сложный запрос с большим ответом.

График (а) демонстрирует зависимость времени обработки запросов с JOIN от количества записей. Наблюдается линейный рост времени обработки с увеличением числа записей до определенного предела, после чего время обработки возрастает более резко. Это указывает на то, что при небольшом и среднем количестве записей СУБД эффективно обрабатывает запросы, но при большом объеме данных возникают проблемы со скоростью обработки из-за увеличения сложности операций соединения.

На графике (б) видно, что время обработки запросов с извлечением больших объектов данных (blob) также увеличивается пропорционально числу записей, однако рост значительно резче, чем при обработке запросов с JOIN. Что может значить, что операции с большими объемами данных заметно увеличивают время обработки запроса, что особенно критично для систем, требующих оперативной обработки информации.

График (в) показывает рост времени обработки заявок при увеличении объема данных в ответе – задержка в БД становится более ощутимой с увеличением количества записей. Это подчеркивает важность оптимизации алгоритмов для обработки «тяжелых» запросов, чтобы обеспечить своевременную обработку и возврат данных.

В целом графики на рис. 6 подтверждают, что с увеличением количества записей в запросе значительно увеличивается время обработки, и это особенно заметно для запросов, требующих сложных операций обработки данных. Данные результаты позволяют сделать вывод о необходимости тщательно планировать ресурсную составляющую СУБД и возможно, использовать специализированные или оптимизированные индексы для ускорения операций поиска и соединения, а также эффективного масштабирования при работе с большими объемами данных.

Заключение

Для построения эффективных систем управления БПЛА, оптимально использующих ресурсы, в том числе связанных серверов СУБД, необходимо иметь возможность оценки параметров реализации запросов к СУБД относительно различных условий, например, при категоризации транзакций, выраженных в виде запросов SQL и особенностей серверных конфигураций.

В статье представлен подход к построению и тестированию многофакторной модели на основе применения имитации. Полученные результаты моделирования учитывают как категоризацию запросов, которые создают неоднородную нагрузку на разные участки и стадии передачи данных, так и виды обработки запросов: сложный запрос с join, сложный запрос с blob, характеризующийся повышенным временем передачи ответа по сети, запрос с большим объемом ответа, сложный запрос с большим ответом и загруженным процессом обработки, запрос за участками карты происходит периодически с установленным периодом, запрос join, выполняющийся через случайные промежутки времени с показательным распределением.

Исходя из представленных данных, можно сделать следующие выводы и предложения по улучшению системы:

1. Система сетевого соединения в большинстве случаев не является узким звеном, однако следует обращать внимание на единичные случаи большого накопления очереди запросов.

2. СУБД может эффективно справляться с большинством запросов, но наличие очереди в БД требует дополнительного рассмотрения, возможно, в виде оптимизации исполнения запросов или масштабирования ресурсов серверов.

3. Общее время обслуживания заявок показывает, что система может быть оптимизирована для предотвращения длительного ожидания ответа для заявок с большой

продолжительностью обработки, возможно, путем оптимизации алгоритмов запросов или улучшения аппаратуры.

4. Принимая во внимание распределение времени обслуживания, важно проанализировать специфику запросов, которые занимают много времени, и выработать стратегии для их оптимизации.

Эти результаты также подчеркивают значимость мониторинга и адаптации систем управления запросами в условиях динамично изменяющейся нагрузки, особенно при использовании БПЛА для мониторинга и сбора данных в реальном времени.

Необходимо учитывать динамику нагрузки на СУБД и применять адаптивные методы работы с запросами для оптимизации производительности, в том числе масштабирование серверов и использование более продвинутых технологий кэширования, индексации и распределенных вычислений.

Настоящее исследование также подчеркивает значимость аналитического подхода к проектированию и эксплуатации БПЛА и СУБД. Предложенные методы и результаты моделирования могут быть использованы для дальнейшего улучшения эффективности систем мониторинга и сбора данных, а также способствовать разработке новых решений для оптимизации работы с большими объемами информации в различных областях применения.

Список источников

1. Акопов А.С. Имитационное моделирование. М.: Изд-во «Юрайт», 2015. 389 с. ISBN 978-5-9916-5549-1. EDN TYRZHT.
2. Доронина Ю.В., Мустафаева М.И. Подход к обработке координат движения беспилотных летательных аппаратов в сельскохозяйственном мониторинге // Известия Тульского государственного университета. Технические науки. 2023. № 4. С. 269–276.
3. Беседин К.Ю., Костенецкий П.С. Моделирование обработки запросов на гибридных вычислительных системах с многоядерными сопроцессорами и графическими ускорителями // Программные системы: теория и приложения. 2014. Т. 5. № 1 (19). С. 91–110.
4. Приказчиков С.О., Костенецкий П.С. Применение графических ускорителей для обработки запросов над сжатыми данными в параллельных системах баз данных // Вестник Южно-Уральского государственного университета. Сер.: Вычислительная математика и информатика. 2015. Т. 4. № 1. С. 64–70.
5. Гасанов Э.Э., Плетнев А.А. Моделирование динамических баз данных // Интеллектуальные системы. Теория и приложения. 2016. Т. 20. № 3. С. 146–150.
6. Тарасов С.В., Бураков В.В. Способы реляционного моделирования иерархических структур данных // Информационно-управляющие системы. 2013. № 6 (67). С. 58–66.
7. Труб И.И. Применение имитационного моделирования к оптимизации индексов баз данных // Девятая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности. 2019. С. 242–248.
8. Телятников А.О. Моделирование и оптимизация распределенных баз данных. 2006.
9. Дубровский А.В. Возможности применения геоинформационного анализа в решении задач мониторинга и моделирования пространственных структур // Известия высших учебных заведений. Геодезия и аэрофотосъемка. 2015. № S5. С. 236–242.
10. Кравченко Ю.А., Лежебоков А.А., Запорожец Д.Ю. Способы интеллектуального анализа данных в сложных системах // Известия Кабардино-Балкарского научного центра РАН. 2012. № 3. С. 52–57.
11. A UAV-cloud system for disaster sensing applications / C. Luo [et al.] // 2015 IEEE 81st Vehicular Technology Conference (VTC Spring). IEEE, 2015. P. 1–5.
12. Erdelj M., Natalizio E. UAV-assisted disaster management: Applications and open issues // 2016 international conference on computing, networking and communications (ICNC). IEEE, 2016. P. 1–5.

13. An autonomous UAV architecture for remote sensing and intelligent decision-making / J. Boubeta-Puig [et al.] // IEEE Internet Computing. 2018. Vol. 22. № 3. P. 6–15.
14. An autonomous UAV architecture for remote sensing and intelligent decision-making / J. Boubeta-Puig [et al.] // IEEE Internet Computing. 2019. Vol. 22. № 5. P. 7–13.
15. An autonomous UAV architecture for remote sensing and intelligent decision-making / J. Boubeta-Puig [et al.] // IEEE Internet Computing. 2018. Vol. 22. № 3. P. 6–15.
16. A small UAV for immediate hazard map generation / R. Hirokawa [et al.] // AIAA Infotech@ Aerospace 2007 Conference and Exhibit. 2007. P. 2725.
17. An efficient architecture for the accurate detection and monitoring of an event through the sky / A. Sharma [et al.] // Computer Communications. 2019. Vol. 148. P. 115–128.
18. Intelligent disaster management system based on cloud-enabled vehicular networks / Z. Alazawi [et al.] // 2011 11th International Conference on ITS Telecommunications. IEEE, 2011. P. 361–368.
19. Itkin M., Kim M., Park Y. Development of cloud-based UAV monitoring and management system // Sensors. 2016. Vol. 16. № 11. P. 1913.
20. Mahmoud S., Mohamed N. Collaborative uavs cloud // 2014 international conference on unmanned aircraft systems (ICUAS). IEEE, 2014. P. 365–373.
21. Help from the sky: Leveraging UAVs for disaster management / M. Erdelj [et al.] // IEEE Pervasive Computing. 2017. Vol. 16. № 1. P. 24–32.
22. Sara M., Jawhar I., Nader M. A softwarization architecture for UAVs and WSNs as Part of the cloud environment // 2016 IEEE international conference on cloud engineering workshop (IC2EW). IEEE, 2016. P. 13–18.
23. Intelligent urban video surveillance system for automatic vehicle detection and tracking in clouds / Y.L. Chen [et al.] // 2013 IEEE 27th international conference on advanced information networking and applications (AINA). IEEE, 2013. P. 814–821.
24. Кобелев Н.Б., Половников В.А., Девятков В.В. Имитационное моделирование. М., 2020.
25. Проворотова А.О., Ширай А.Е., Гайденко Т.И. Особенности современного сервера баз данных // Системы компьютерной математики и их приложения: сб. статей. С. 118.
26. Мухаметшин, Р. М. Развертывание сервера баз данных // Научные исследования: проблемы и перспективы: сб. науч. трудов по материалам XXXVII Междунар. науч.-практ. конф. Анапа: Науч.-исслед. центр эконом. и соц. процессов, 2022. С. 134–138. EDN VTQAVG.
27. Григорьев Ю.А. 77-30569/296127 Организация базы данных в программном комплексе анализа характеристик производительности распределённых систем обработки данных // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2012. № 2. С. 39–39.

References

1. Akopov A.S. Imitacionnoe modelirovanie. M.: Izd-vo «Yurajt», 2015. 389 s. ISBN 978-5-9916-5549-1. EDN TYPZHT.
2. Doronina Yu.V., Mustafaeva M.I. Podhod k obrabotke koordinat dvizheniya bespilotnyh letatel'nyh apparatov v sel'skohozyajstvennom monitoringe // Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskie nauki. 2023. № 4. S. 269–276.
3. Besedin K.Yu., Kosteneckij P.S. Modelirovanie obrabotki zaprosov na gibridnyh vychislitel'nyh sistemah s mnogoyadernymi soprocessorami i graficheskimi uskoritelyami // Programmnye sistemy: teoriya i prilozheniya. 2014. T. 5. № 1 (19). S. 91–110.
4. Prikazchikov S.O., Kosteneckij P.S. Primenenie graficheskikh uskoritelej dlya obrabotki zaprosov nad szhatymi dannymi v parallel'nyh sistemah baz dannyh // Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Ser.: Vychislitel'naya matematika i informatika. 2015. T. 4. № 1. S. 64–70.
5. Gasanov E.E., Pletnev A.A. Modelirovanie dinamicheskikh baz dannyh // Intellektual'nye sistemy. Teoriya i prilozheniya. 2016. T. 20. № 3. S. 146–150.
6. Tarasov S.V., Burakov V.V. Sposoby relyacionnogo modelirovaniya ierarhicheskikh struktur dannyh // Informacionno-upravlyayushchie sistemy. 2013. № 6 (67). S. 58–66.

7. Trub I.I. *Primenenie imitacionnogo modelirovaniya k optimizacii indeksov baz dannyh* // *Devyataya vserossijskaya nauchno-prakticheskaya konferenciya po imitacionnomu modelirovaniyu i ego primeneniyu v nauke i promyshlennosti*. 2019. S. 242–248.
8. Telyatnikov A.O. *Modelirovanie i optimizaciya raspredelennyh baz dannyh*. 2006.
9. Dubrovskij A.V. *Vozmozhnosti primeneniya geoinformacionnogo analiza v reshenii zadach monitoringa i modelirovaniya prostranstvennyh struktur* // *Izvestiya vysshih uchebnyh zavedenij. Geodeziya i aerofotos"emka*. 2015. № S5. S. 236–242.
10. Kravchenko Yu.A., Lezhebokov A.A., Zaporozhec D.Yu. *Sposoby intellektual'nogo analiza dannyh v slozhnyh sistemah* // *Izvestiya Kabardino-Balkarskogo nauchnogo centra RAN*. 2012. № 3. S. 52–57.
11. *A UAV-cloud system for disaster sensing applications* / C. Luo [et al.] // *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015. P. 1–5.
12. Erdelj M., Natalizio E. *UAV-assisted disaster management: Applications and open issues* // *2016 international conference on computing, networking and communications (ICNC)*. IEEE, 2016. P. 1–5.
13. *An autonomous UAV architecture for remote sensing and intelligent decision-making* / J. Boubeta-Puig [et al.] // *IEEE Internet Computing*. 2018. Vol. 22. № 3. P. 6–15.
14. *An autonomous UAV architecture for remote sensing and intelligent decision-making* / J. Boubeta-Puig [et al.] // *IEEE Internet Computing*. 2019. Vol. 22. № 5. P. 7–13.
15. *An autonomous UAV architecture for remote sensing and intelligent decision-making* / J. Boubeta-Puig [et al.] // *IEEE Internet Computing*. 2018. Vol. 22. № 3. P. 6–15.
16. *A small UAV for immediate hazard map generation* / R. Hirokawa [et al.] // *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*. 2007. P. 2725.
17. *An efficient architecture for the accurate detection and monitoring of an event through the sky* / A. Sharma [et al.] // *Computer Communications*. 2019. Vol. 148. P. 115–128.
18. *Intelligent disaster management system based on cloud-enabled vehicular networks* / Z. Alazawi [et al.] // *2011 11th International Conference on ITS Telecommunications*. IEEE, 2011. P. 361–368.
19. Itkin M., Kim M., Park Y. *Development of cloud-based UAV monitoring and management system* // *Sensors*. 2016. Vol. 16. № 11. P. 1913.
20. Mahmoud S., Mohamed N. *Collaborative uavs cloud* // *2014 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2014. P. 365–373.
21. *Help from the sky: Leveraging UAVs for disaster management* / M. Erdelj [et al.] // *IEEE Pervasive Computing*. 2017. Vol. 16. № 1. P. 24–32.
22. Sara M., Jawhar I., Nader M. *A softwarization architecture for UAVs and WSNs as Part of the cloud environment* // *2016 IEEE international conference on cloud engineering workshop (IC2EW)*. IEEE, 2016. P. 13–18.
23. *Intelligent urban video surveillance system for automatic vehicle detection and tracking in clouds* / Y.L. Chen [et al.] // *2013 IEEE 27th international conference on advanced information networking and applications (AINA)*. IEEE, 2013. P. 814–821.
24. Kobelev N.B., Polovnikov V.A., Devyatkov V.V. *Imitacionnoe modelirovanie*. M., 2020.
25. Provorotova A.O., Shiraj A.E., Gajdenko T.I. *Osobennosti sovremennogo servera baz dannyh* // *Sistemy komp'yuternoj matematiki i ih prilozheniya: sb. statej*. S. 118.
26. Muhametshin, R. M. *Razvertyvanie servera baz dannyh* // *Nauchnye issledovaniya: problemy i perspektivy: sb. nauch. trudov po materialam XXXVII Mezhdunar. nauch.-prakt. konf. Anapa: Nauch.-issled. centr ekonom. i soc. processov*, 2022. S. 134–138. EDN VTQAVG.
27. Grigor'ev Yu.A. *77-30569/296127 Organizaciya bazy dannyh v programmnom komplekse analiza karakteristik proizvoditel'nosti raspredelyonnyh sistem obrabotki dannyh* // *Nauka i obrazovanie: nauchnoe izdanie MGTU im. N.E. Baumana*. 2012. № 2. S. 39–39.

Информация о статье:

Статья поступила в редакцию: 21.02.2024; одобрена после рецензирования: 04.03.2024;
принята к публикации: 06.03.2024

The information about article:

The article was submitted to the editorial office: 21.02.2024; approved after review: 04.03.2024;
accepted for publication: 06.03.2024

Информация об авторах:

Доронина Юлия Валентиновна, профессор кафедры «Информационные технологии и компьютерные системы» Севастопольского государственного университета (299053, г. Севастополь, ул. Университетская, д. 33), доктор технических наук, доцент, e-mail: YVDoronina@sevsu.ru, SPIN-код: 8548-6746

Мустафаева Мерьем Ибраимовна, аспирант кафедры «Информационные технологии и компьютерные системы» Севастопольского государственного университета (299053, г. Севастополь, ул. Университетская, д. 33)

Information about authors:

Doronina Yulia V., professor of the department of «Information technologies and computer systems» Sevastopol state university (299053, Sevastopol, st. Universitetskaya, 33), doctor of technical sciences, associate professor, e-mail: YVDoronina@sevsu.ru, SPIN: 8548-6746

Mustafaeva Meryem I., graduate student of the department of «Information technologies and computer systems» Sevastopol state university (299053, Sevastopol, st. Universitetskaya, 33)