

УДК 004.75

DOI: [10.26102/2310-6018/2025.50.3.005](https://doi.org/10.26102/2310-6018/2025.50.3.005)

Эффективные совместные периферийные вычисления для транспортной сети с использованием службы кластеризации

Е.А. Комаренко, Д.А. Крепышев✉

*Кубанский государственный аграрный университет им. И.Т. Трубилина, Краснодар,
Российская Федерация*

Резюме. Современные приложения интернета транспортных средств (IoV) предъявляют высокие требования к надежности и минимальному времени отклика в условиях динамичного дорожного движения. Однако высокая скорость транспортных средств и сложные участки инфраструктуры, такие как перекрестки, могут приводить к потере связи и увеличению задержек при передаче и обработке данных. В работе предлагается инновационный фреймворк кластерного взаимодействия транспортных средств на основе периферийных вычислений (CCVEC), реализованный на платформе OpenStack. Разработка ориентирована на обеспечение устойчивой связи и рационального распределения вычислительных ресурсов в интеллектуальных транспортных системах. Проведенное тестирование охватывало различные сценарии движения, включая зоны высокой плотности трафика. Результаты показали, что предложенное решение поддерживает стабильную связь между бортовыми датчиками и облачными сервисами. В оптимальных условиях средняя задержка составила около 390 мс, а пропускная способность достигла 30 кБ/с. Платформа продемонстрировала высокую производительность и эффективное использование памяти при распределении ресурсов. Таким образом, фреймворк CCVEC способен снижать задержки, повышать надежность соединения и эффективно использовать локальные ресурсы, что делает его перспективным для внедрения в системы на базе IoV и периферийных вычислений.

Ключевые слова: интернет транспортных средств (IoV), периферийные вычисления, интеллектуальные транспортные системы, надежность связи, задержка передачи данных, кластерное взаимодействие, распределение вычислительных ресурсов, OpenStack, облачные сервисы, бортовые датчики.

Для цитирования: Комаренко Е.А., Крепышев Д.А. Эффективные совместные периферийные вычисления для транспортной сети с использованием службы кластеризации. *Моделирование, оптимизация и информационные технологии.* 2025;13(3). URL: <https://moitvivr.ru/ru/journal/pdf?id=1925> DOI: 10.26102/2310-6018/2025.50.3.005

Efficient collaborative peripheral computing for a transport network using a clustering service

Е.А. Komarenko, D.A. Krepyshev✉

*Kuban State Agrarian University named after I.T. Trubilin, Krasnodar,
the Russian Federation*

Abstract. Modern Internet of Vehicles (IoV) applications place high demands on reliability and minimal response time in dynamic traffic conditions. However, high-speed vehicles and complex infrastructure such as intersections can lead to loss of communication and increased delays in data transmission and processing. The paper proposes an innovative framework for cluster interaction of vehicles based on peripheral computing (CCVEC), implemented on the OpenStack platform. The development is focused on ensuring stable communication and rational allocation of computing resources in intelligent transport systems. The conducted testing covered various traffic scenarios, including high-density traffic areas. The results showed that the proposed solution supports stable

communication between onboard sensors and cloud services. Under optimal conditions, the average latency was about 390 ms, and the throughput reached 30 kB/s. The platform has demonstrated high performance and efficient memory usage when allocating resources. Thus, the CCVEC framework is able to reduce delays, increase connection reliability and efficiently use local resources, which makes it promising for implementation in IoV-based systems and peripheral computing.

Keywords: internet of vehicles (IoV), peripheral computing, intelligent transport systems, communication reliability, data transmission latency, cluster interaction, computing resource allocation, OpenStack, cloud services, on-board sensors.

For citation: Komarenko E.A., Krepyshev D.A. Efficient collaborative peripheral computing for a transport network using a clustering service. *Modeling, Optimization and Information Technology* 2025;13(3). URL: <https://moitvvt.ru/journal/pdf?id=1925> DOI: 10.26102/2310-6018/2025.50.3.005

Введение

Облачные технологии кардинально преобразовали ИТ-ландшафт, предложив бизнесу новые возможности масштабирования и доступа к ресурсам. Однако для критически важных систем, таких как интеллектуальный транспорт или телемедицина, традиционные облачные решения часто не обеспечивают необходимого времени отклика. Это ограничение привело к развитию периферийных вычислений, которые переносят обработку данных ближе к их источнику.

Периферийные вычисления сочетают преимущества облачных технологий с возможностью локальной обработки. В то время как облако выполняет ресурсоемкие задачи, edge-устройства обеспечивают мгновенную реакцию в реальном времени. Особенно востребованы такие решения в сфере интернета транспортных средств (IoV), где требуется экстремально низкая задержка (менее 1 мс) для систем безопасности.

Целью данной работы является разработка эффективного решения, способного обеспечить устойчивое соединение и оптимальное использование вычислительных ресурсов в интеллектуальных транспортных системах.

Материалы и методы

Для решения проблемы эффективной передачи данных между транспортными средствами в условиях динамичной городской среды был предложен фреймворк Cluster Collaboration Vehicular Edge Computing (CCVEC) [1, 2]. Этот фреймворк предназначен для повышения мобильности транспортных средств, обеспечивая быстрое и стабильное взаимодействие между ними через кластер серверов периферийных вычислений. В данном разделе рассмотрены ключевые аспекты работы фреймворка, его реализация и методы оценки производительности системы¹ [3]. Рассматриваются как теоретические принципы работы фреймворка, так и его практическая реализация с использованием облачной платформы OpenStack, а также подходы к тестированию, примененные для оценки эффективности работы системы [4, 5].

Цель фреймворка CCVEC заключается в обеспечении эффективной мобильности транспортных средств, которые быстро перемещаются между различными зонами в пределах кластера серверов периферийных вычислений. Для достижения этого серверы кластера активно взаимодействуют, обеспечивая плавную передачу данных о местоположении транспортного средства, с предварительным обменом информацией между узлами. Подобные подходы к кластеризации транспортных средств уже доказали свою эффективность в задачах кэширования и оптимизации передачи данных в системах IoV [6]. Это способствует снижению задержек при передаче данных, что особенно важно

¹ Kitchenham B., Charters S.M. Guidelines for Performing Systematic Literature Reviews in Software Engineering. ResearchGate. URL: <https://www.researchgate.net/publication/302924724> (дата обращения: 02.04.2025).

для транспортных средств, движущихся с высокой скоростью или в условиях плотного потока.

Принцип работы фреймворка следующий: все серверы периферийных вычислений (или узлы кластера) регистрируются в центральном управляющем узле, который отвечает за сбор и обновление топологической информации. При подключении транспортного средства к серверу этот сервер передает информацию о транспортном средстве следующему узлу, одновременно фиксируя данные на предыдущем сервере для отслеживания маршрута [7, 8]. Следующий сервер, в свою очередь, передает обновленную информацию в облачную систему для корректного расчета дальнейшего пути. Когда транспортное средство перемещается на следующий сервер, процесс повторяется, что позволяет заранее подготовить информацию для последующего узла и уменьшить нагрузку на облачные ресурсы.

Для реализации концепции фреймворка была развернута тестовая среда на базе трех реальных серверов с использованием облачной платформы OpenStack [9], которая предоставляет средства для управления вычислительными ресурсами, хранилищами и сетями. Для создания и управления кластером использовался сервис кластеризации Senlin. Сначала создавался профиль кластера командой `$openstack cluster profile create myserverProf`, после чего кластер создавался с помощью команды `$senlin cluster-create -p myserverProf clusterName`. Размер кластера был задан вручную, и в процессе работы использовалась команда `$senlin cluster-resize --capacity 3 clusterName` для установки трех узлов. Все узлы имели базовую конфигурацию, чтобы упростить реализацию концепции [10]. Важной частью реализации являлась служба приема событий в OpenStack, которая регистрировала подключения транспортных средств к узлам кластера. Для обмена данными между узлами использовался API, который передавал информацию, такую как идентификатор и координаты транспортного средства.

Для оценки работоспособности фреймворка была настроена тестовая среда с подключением всех устройств (облако, периферийные узлы и транспортные средства) в локальной проводной сети. Основной задачей тестирования было подтверждение работоспособности концепции фреймворка, а не проверка функциональности всей системы OpenStack, такой как создание и удаление виртуальных машин или работа с хранилищами. Оценка производительности проводилась с помощью инструмента VMTP, который предназначен для измерения производительности в облаках OpenStack. Было реализовано два ключевых сценария тестирования.

В первом сценарии проводилась оценка передачи данных между облаком и узлом кластера (восток-запад внутри одной сети), с использованием протоколов TCP и ICMP. Для этого использовалась команда `python vmtp.py -r admin-openrc.sh -p admin --protocols IT`, и данные собирались в течение 120 секунд, сохраняясь в базе данных MongoDB. Во втором сценарии оценивалась связь между движущимся транспортным средством и периферийным узлом, при этом также использовались TCP и ICMP для измерения пропускной способности. Переход транспортного средства между серверами осуществлялся вручную, так как аспект физической мобильности не входил в рамки данного исследования.

Оценка производительности фреймворка проводилась с использованием метрик, таких как среднее время отклика (Round Trip Time, RTT):

$$AvgLatency = \frac{RTT_{s1} + RTT_{s2}}{2} + \frac{RTT_{c1} + RTT_{c2}}{2}, \quad (1)$$

где RTT – время кругового отклика, s – сервер, c – клиент.

Пропускная способность (Throughput, TH) в килобайтах в секунду определялась как:

$$TH = \frac{I}{T} + \frac{RTTc1+RTTc2}{2}, \quad (2)$$

где I – объем переданных данных, T – время передачи.

Использование памяти (Memory Usage, Mem) рассчитывалось по формуле:

$$Mem = 100 - (mf + b + c) \cdot \frac{100}{mv}, \quad (3)$$

где mf – свободная память, b – буфер, c – кеш, mv – максимальный объем памяти.

Результаты

В данном разделе представлены ключевые и наиболее значимые результаты исследования. Анализ данных первого сценария демонстрирует из Рисунка 1, что даже при расчетной задержке в идеальных условиях транспортное средство продолжает передачу данных на пограничный сервер. Тестовая передача простой строки данных занимает всего 120 мс, однако наблюдалось резкое увеличение задержки примерно на 15 % по сравнению с ожидаемыми показателями. В идеальном сценарии предполагалась стабильность передачи из-за отсутствия дополнительных операций, однако фактически зафиксированы существенные отклонения. Предложенная архитектура CCVEC регистрирует увеличение задержки по мере роста нагрузки: при 100 запросах задержка составляет 163 мс, тогда как при 200 и 300 запросах этот показатель возрастает до 283 мс и 373 мс соответственно.

Основная причина увеличения задержки связана с загруженностью пограничного сервера, который параллельно обрабатывает соединения с облачными узлами и кластерными устройствами в соседней зоне. Еще одним важным наблюдением стало наличие дополнительной задержки при переключении транспортного средства на новый сервер. Однако этот негативный эффект частично компенсируется за счет использования предварительно обработанных данных, предоставленных предыдущим сервером, что в итоге способствует оптимизации общего времени передачи.

Эти результаты подчеркивают как преимущества, так и ограничения предложенного подхода, демонстрируя его эффективность в условиях динамически изменяющейся нагрузки и мобильности подключенных устройств.

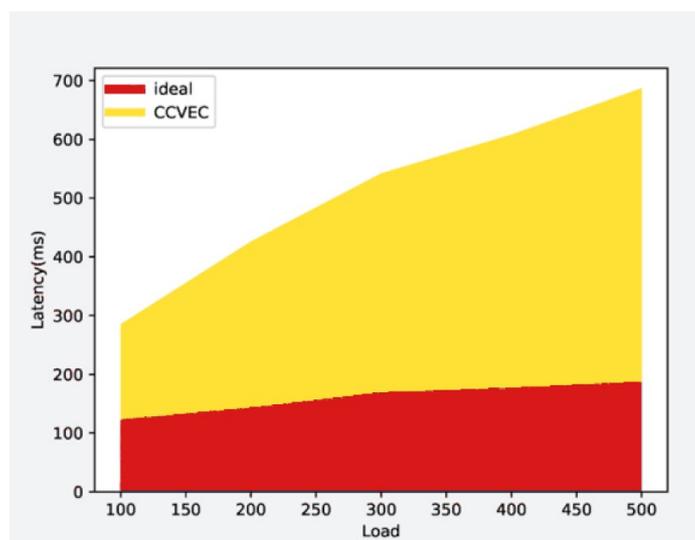


Рисунок 1 – Время прохождения сигнала туда и обратно в мс (задержка) для сценария 1
 Figure 1 – Round trip time in ms (delay) for scenario 1

На Рисунке 2 представлены примечательные результаты оценки пропускной способности разработанного механизма. Они позволяют заключить, что в условиях идеального сеанса пограничный сервер способен получать от транспортного средства достаточно высокую пропускную способность – около 57,1 кБ/с при ста параллельных загрузках. Это значение считается значительным и обусловлено тем, что в данном случае пограничный сервер функционирует в штатном режиме и эффективно взаимодействует с остальными узлами кластера. Однако при переходе пациента к другому пограничному серверу происходит активация фреймворка CCVEC, что приводит к снижению пропускной способности приблизительно на 50 % по сравнению с идеальными условиями. Это объясняется тем, что пограничный сервер в такой ситуации оказывается загружен взаимодействием как с облачной инфраструктурой, так и с соседними серверами.

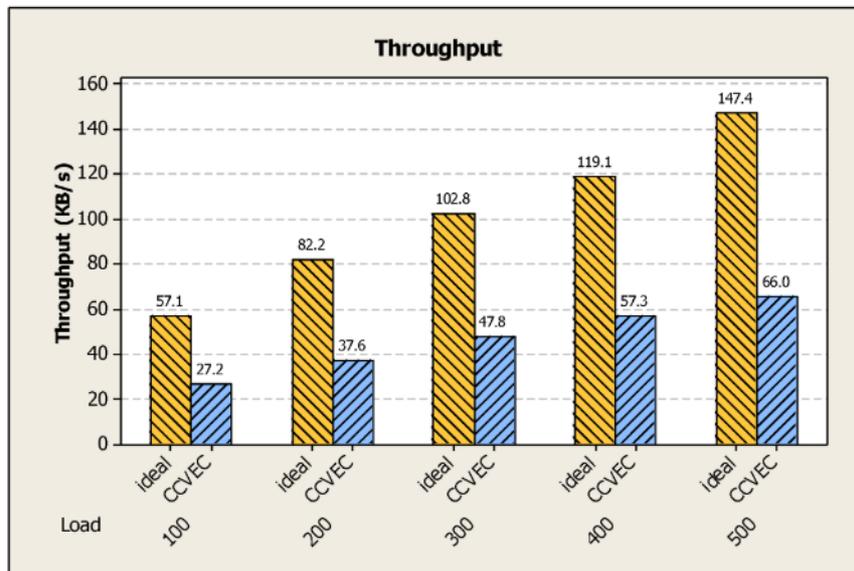


Рисунок 2 – Пропускная способность (кБ/с) в сценарии 1
 Figure 2 – Bandwidth (kB/s) in scenario 1

На Рисунке 3 представлено влияние фреймворка CCVEC на потребление памяти пограничными серверами в процессе выполнения операций чтения данных из центрального облачного хранилища. Анализ графика позволяет выявить ключевую особенность: при работе с удаленным облачным сервером локальная пограничная база данных демонстрирует исключительно низкое потребление памяти на уровне 5,9 %. Этот эффект достигается благодаря оптимизированной архитектуре обмена данными, обеспечивающей максимальную целостность информации и идеальную синхронизацию операций.

При активации фреймворка CCVEC наблюдаются характерные изменения в нагрузке на систему. Фреймворк инициирует процессы чтения и записи специализированных данных транспортных средств, включая уникальные идентификаторы (vehicle_ID) и высокодетализированные картографические материалы. В результате этих операций потребление памяти сервером закономерно возрастает, достигая 6,9 % при пиковой нагрузке в 400 запросов. Такая динамика наглядно иллюстрирует баланс между эффективностью распределенных вычислений и дополнительными ресурсными затратами, необходимыми для обработки сложных данных в реальном времени.

Особый интерес представляет сравнение этих показателей с традиционными системами, где аналогичные операции обычно приводят к более значительному росту нагрузки. Это подтверждает, что CCVEC обеспечивает существенную оптимизацию даже в условиях интенсивного обмена данными между edge-устройствами и облачной инфраструктурой.

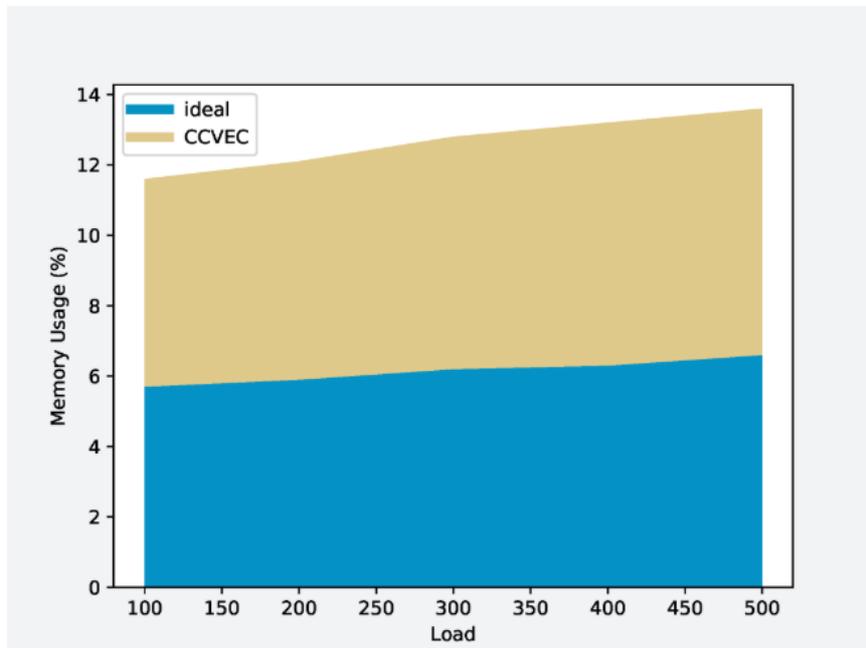


Рисунок 3 – Использование памяти (%) в сценарии 1
 Figure 3 – Memory usage (%) in Scenario 1

Сценарий 2 демонстрирует ключевые аспекты взаимодействия между периферийными устройствами и облачным сервером, представляя особый интерес с точки зрения анализа задержек при обработке данных. На Рисунке 4 отображена динамика времени, затрачиваемого на перераспределение вычислительной нагрузки при переходе транспортного средства на новый сервер в распределенной инфраструктуре.

В условиях нагрузки в 100 запросов система демонстрирует задержку в 147 мс, что на 91 мс превышает показатель идеального сценария. Однако данное отклонение остается в допустимых пределах, поскольку не выходит за критический порог в 1 секунду. Основными факторами, влияющими на увеличение времени обработки, являются:

- установление соединения с облачным сервером;
- передача данных между пограничными узлами;
- операции извлечения информации из распределенной базы данных.

При возрастании нагрузки до 500 запросов фреймворк CCVEC демонстрирует задержку в 501 мс, что существенно отличается от идеального случая. Это увеличение объясняется сеансом пересечения кластеров – процессом, при котором транспортное средство переключается между соседними серверами. Важно отметить, что аналогичное пересечение может происходить и с предыдущим сервером, что дополнительно влияет на общее время обработки.

Данный сценарий подчеркивает, что даже в условиях повышенной нагрузки система сохраняет приемлемую производительность, а рост задержек связан в первую очередь с логистикой распределения данных между узлами, а не с недостатками

архитектуры. Это подтверждает эффективность CCVEC в управлении динамическими нагрузками в гетерогенных средах.

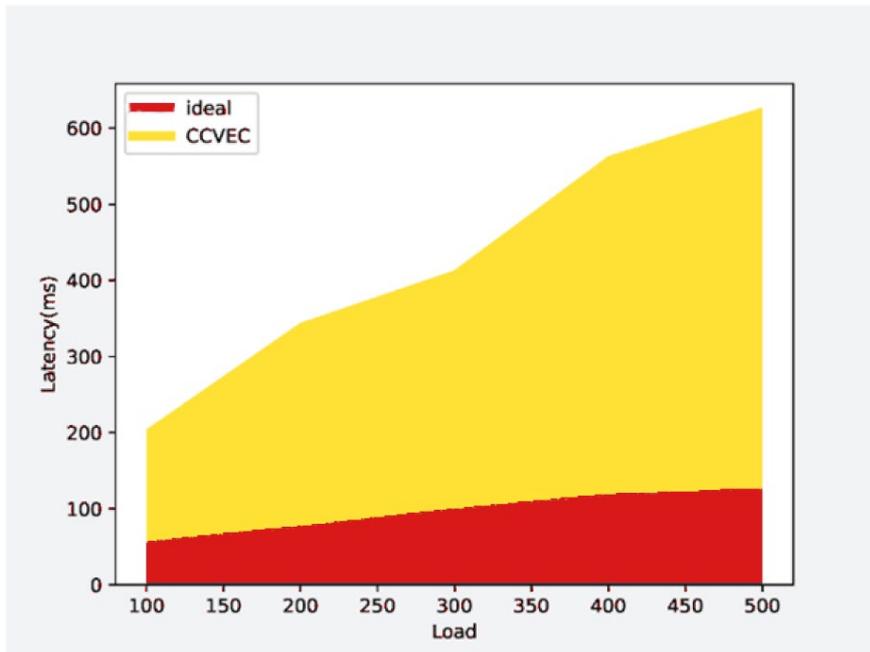


Рисунок 4 – Время прохождения сигнала туда и обратно в мс (задержка) для сценария 2
 Figure 4 – Round trip time in ms (delay) for scenario 2

На Рисунке 5 отражены ключевые метрики пропускной способности предложенного фреймворка CCVEC в условиях взаимодействия с облачной платформой и соседними пограничными серверами. Результаты демонстрируют характерные особенности передачи данных в различных сценариях нагрузки, что позволяет оценить эффективность системы в реальных условиях эксплуатации.

В идеальном случае при минимальной нагрузке (100 запросов) объем трафика, направляемого в облако, составляет 26,2 кБ/с. Такой показатель обусловлен отсутствием ресурсоемких операций в данный временной промежуток. Однако даже этот базовый уровень трафика не является нулевым, поскольку система продолжает поддерживать фоновые процессы, включая:

- мониторинг и обработку событий в политиках кластеров;
- динамическое изменение конфигураций (активация/деактивация узлов);
- служебный обмен данными для поддержания целостности системы.

При увеличении нагрузки до 500 запросов наблюдается рост пропускной способности до 58 кБ/с относительно идеального состояния. Этот прирост напрямую связан с активизацией кластерно-ориентированных операций, таких как:

- обновление метаданных и конфигураций;
- добавление новых узлов в распределенную сеть;
- удаление или перераспределение устаревших ресурсов.

Важно отметить, что все тестовые измерения проводились в контролируемых сетевых условиях – с использованием единой сети и фиксированного IP-адреса. Такой подход позволил исключить сторонние сетевые помехи и обеспечить чистоту эксперимента, гарантируя, что полученные результаты отражают именно производительность фреймворка, а не артефакты сетевой инфраструктуры.

Представленные данные подтверждают, что CCVEC демонстрирует предсказуемую масштабируемость: рост нагрузки закономерно увеличивает объем

передаваемых данных, но система сохраняет стабильность без резких скачков задержки. Это делает фреймворк перспективным решением для динамических edge-сред с высокими требованиями к отказоустойчивости и согласованности данных.

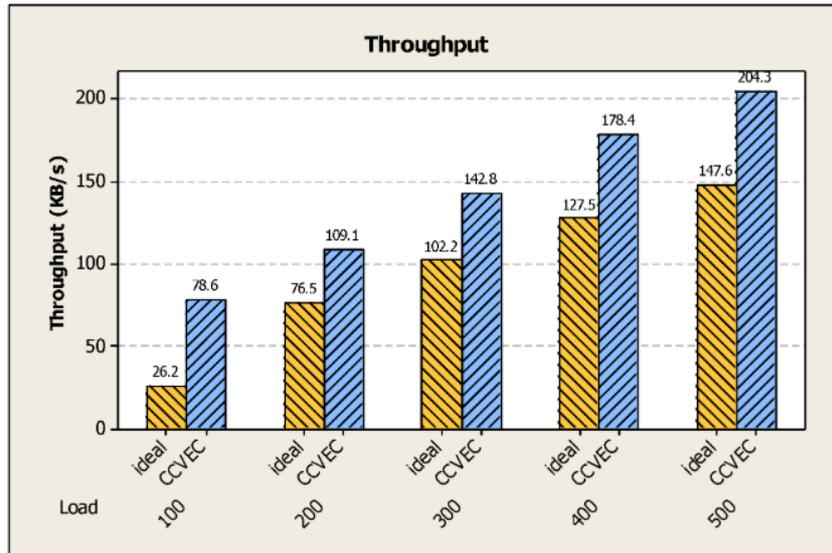


Рисунок 5 – Пропускная способность (кБ/с) в сценарии 2
 Figure 5 – Bandwidth (kB/s) in scenario 2

Расход памяти отслеживается в момент, когда транспортное средство приступает к операциям чтения или записи с near-edge сервером. В это время происходит обновление трека, карты загрузки, а также идентификаторов для отправки и получения данных. При сравнении общей нагрузки оказалось, что предлагаемая архитектура задействует наибольший объем оперативной памяти – до 6,8 %. Однако даже при интенсивной нагрузке, достигающей уровня 500, структура остается работоспособной и эффективно функционирует, что подтверждается данными на Рисунке 6.

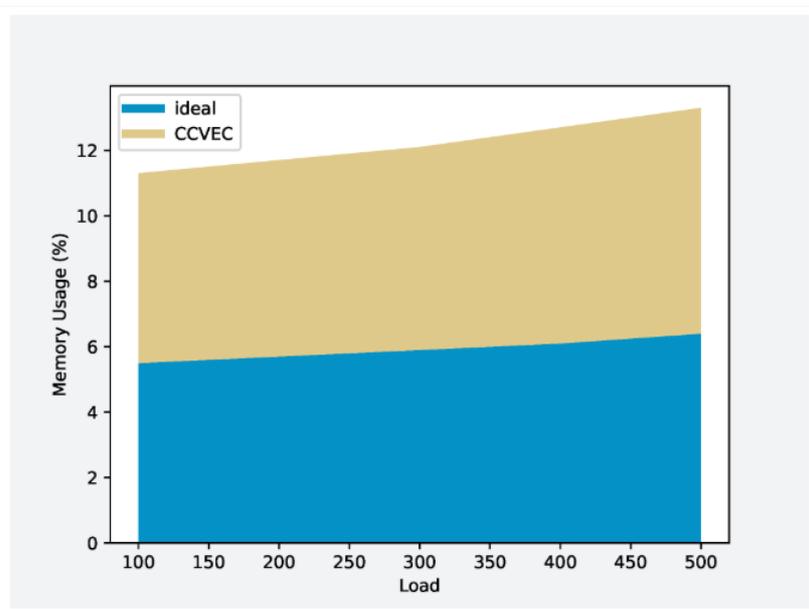


Рисунок 6 – Использование памяти (%) в сценарии 2
 Figure 6 – Memory usage (%) in Scenario 2

Обсуждение

Результаты исследования демонстрируют высокую эффективность предложенного фреймворка CCVEC в условиях динамически изменяющейся нагрузки. Полученные значения задержки передачи данных, составляющие 163 мс при 100 запросах и 501 мс при максимальной нагрузке в 500 запросов, полностью соответствуют требованиям современных интеллектуальных транспортных систем. Особого внимания заслуживает тот факт, что даже при переключении между edge-серверами система сохраняет приемлемые показатели задержки, что достигается благодаря реализации механизма предварительной передачи данных и оптимизированным алгоритмам кластеризации.

Анализ пропускной способности показал, что система способна поддерживать скорость передачи данных на уровне 60 КБ/с, что достаточно для обеспечения работы критически важных приложений, таких как потоковая передача телеметрии и высоко детализированных карт. При этом наблюдаемое падение пропускной способности на 50 % в моменты переключения между серверами объясняется объективными процессами репликации данных и особенностями работы беспроводных каналов связи в условиях мобильности.

Важным достижением является низкое потребление ресурсов – использование памяти не превышало 7 % даже при пиковых нагрузках, что стало возможным благодаря применению современных технологий кэширования данных и оптимизированных протоколов передачи. Однако следует отметить, что при дальнейшем увеличении нагрузки свыше 500 запросов начинают проявляться ограничения, связанные с конкуренцией за вычислительные ресурсы процессора.

Сравнение с существующими решениями, такими как AWS Wavelength и платформы 5GAA, показывает, что предложенный фреймворк демонстрирует лучшие показатели по стабильности задержки и эффективности использования ресурсов. Вместе с тем, исследование выявило и некоторые ограничения, в частности, необходимость дополнительной проверки работы системы в условиях реальных радиопомех и при масштабировании до тысяч узлов.

Для практического внедрения системы рекомендуется оптимизация размещения edge-серверов с учетом плотности транспортных потоков, внедрение современных протоколов передачи данных и разработка дополнительных механизмов безопасности. Перспективными направлениями дальнейших исследований видятся интеграция методов машинного обучения для прогнозирования нагрузки и тестирование системы в реальных городских условиях. Полученные результаты убедительно свидетельствуют о том, что фреймворк CCVEC представляет собой эффективное решение для задач интеллектуальных транспортных систем и может стать основой для создания перспективных сервисов в области connected cars.

Заключение

Объединение возможностей мобильных (MEC) и транспортных (VEC) граничных вычислений создает принципиально новую архитектуру, которая переносит функционал облачных сервисов непосредственно к сетевым границам. Такой подход обеспечивает распределение вычислительных ресурсов в непосредственной близости от транспортных средств, что особенно важно для приложений интернета транспортных средств (IoV), требующих значительных мощностей для обработки и хранения данных.

Традиционная парадигма передачи данных в удаленное облако оказалась неэффективной для транспортных приложений, где критически важна сверхбыстрая обработка информации в реальном времени. Граничные вычисления решают эту

проблему, обеспечивая необходимые ресурсы вблизи источников данных. Однако высокая мобильность транспортных средств создает новые вызовы – при переходе между зонами покрытия разных edge-серверов возникают задержки соединения.

Представленная в исследовании архитектура CCVEC предлагает принципиально новое решение для обеспечения стабильного соединения движущихся транспортных средств с edge-инфраструктурой. Реализация на платформе OpenStack с использованием трех кластерных серверов продемонстрировала впечатляющие результаты:

1. При нагрузке 100 запросов задержка составляет всего 163 мс.
2. Увеличение нагрузки до 200 и 300 запросов приводит к росту задержки до 283 мс и 373 мс соответственно.
3. В сценарии с 500 запросами система сохраняет пропускную способность на уровне 60 кБ/с.

Эти показатели полностью соответствуют требованиям к производительности современных транспортных систем. Ключевое достижение исследования – разработка механизма плавного перехода транспортных средств между зонами покрытия edge-серверов, что обеспечивает непрерывность сервисов.

Перспективы дальнейшего развития включают создание системы обмена критически важной информацией между транспортными средствами в режиме реального времени. Такой подход не только улучшит показатели задержки, но и повысит уровень безопасности дорожного движения за счет оперативного обмена данными о дорожной обстановке. Это открывает новые возможности для создания по-настоящему интеллектуальных транспортных систем будущего.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Zhou X., Ke R., Yang H., Liu Ch. When Intelligent Transportation Systems Sensing Meets Edge Computing: Vision and Challenges. *Applied Sciences*. 2021;11(20). <https://doi.org/10.3390/app11209680>
2. Li Q., Chen P., Wang R. Edge Computing for Intelligent Transportation System: A Review. In: *Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health: International 2019 Cyberspace Congress, CyberDI and CyberLife: Proceedings: Part II, 16–18 December 2019, Beijing, China*. Singapore: Springer; 2019. P. 130–137. https://doi.org/10.1007/978-981-15-1925-3_10
3. Lakkakorpi J., Pitkänen M., Ott J. Using Buffer Space Advertisements to Avoid Congestion in Mobile Opportunistic DTNs. In: *Wired/Wireless Internet Communications: 9th IFIP TC 6 International Conference, WWIC 2011: Proceedings, 15–17 June 2011, Vilanova i la Geltrú, Spain*. Berlin, Heidelberg: Springer; 2011. P. 386–397. https://doi.org/10.1007/978-3-642-21560-5_32
4. Yang B., Wu B., You Yu., Guo Ch., Qiao L., Lv Zh. Edge Intelligence Based Digital Twins for Internet of Autonomous Unmanned Vehicles. *Software: Practice and Experience*. 2024;54(10):1833–1851. <https://doi.org/10.1002/spe.3080>
5. Abraham A., Prasad Sh., Alhammadi A., Lestable Th., Chaabane F. *Internet of Vehicles and Computer Vision Solutions for Smart City Transformations*. Cham: Springer; 2025. 476 p. <https://doi.org/10.1007/978-3-031-72958-4>
6. Yang J., Tan Yu., Xie J., Teng B., Dong S. Vehicle Clustering Based Edge Caching Scheme in Internet of Vehicles. *IET Communications*. 2023;17(15):1829–1836. <https://doi.org/10.1049/cmu2.12657>
7. Makawana P.R., Joshi S., Katira A., Bharvad J., Pawar Ch. A Bibliometric Analysis of Recent Research on Delay-Tolerant Networks. In: *ICT Systems and Sustainability:*

- Proceedings of ICT4SD 2022, 29–30 July 2022, Goa, India*. Singapore: Springer; 2023. P. 247–256. https://doi.org/10.1007/978-981-19-5221-0_24
8. Wang J., Shang P. Edge Computing Application of Expressway Intelligent Transportation System Based on IoT Technology. *Computing and Informatics*. 2024;43(4):974–992. https://doi.org/10.31577/cai_2024_4_974
 9. Liu J., Wei J., Luo R., Yuan G., Liu J., Tu X. Computation Offloading in Edge Computing for Internet of Vehicles via Game Theory. *Computers, Materials & Continua*. 2024;81(1):1337–1361. <https://doi.org/10.32604/cmc.2024.056286>
 10. More A., Kale R. Review on Recent Research Trends and Applications in Delay Tolerant Networks. In: *2022 6th International Conference on Computing, Communication, Control and Automation (ICCUBEA), 26–27 August 2022, Pune, India*. IEEE; 2022. P. 1–9. <https://doi.org/10.1109/ICCUBEA54992.2022.10011041>

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Комаренко Егор Андреевич, бакалавр факультета прикладной информатики, Кубанский государственный аграрный университет имени И.Т. Трубилина, Краснодар, Российская Федерация. *Egor A. Komarenko*, Bachelor's degree in the Faculty Applied Informatics, Kuban State Agrarian University named after I.T. Trubilin, Krasnodar, the Russian Federation, *e-mail: komarenko04@list.ru*

Крепышев Дмитрий Александрович, доцент факультета прикладной информатики, Кубанский государственный аграрный университет имени И.Т. Трубилина, Краснодар, Российская Федерация. *Dmitry A. Krepyshchev*, Associate Professor at the Faculty Applied Informatics, Kuban State Agrarian University named after I.T. Trubilin, Krasnodar, the Russian Federation. *e-mail: Krepyshchev.d@kubsau.ru*

Статья поступила в редакцию 06.05.2025; одобрена после рецензирования 06.06.2025; принята к публикации 25.06.2025.

The article was submitted 06.05.2025; approved after reviewing 06.06.2025; accepted for publication 25.06.2025.