

УДК 621.396.41

Е.В. Коршунова, И.С. Сыркин, В.Ю. Садовец

## Архитектурное сравнение ROS1 и ROS2 в контексте работы с данными облаков точек лидара

Проводится сравнительный анализ архитектурных особенностей ROS1 и ROS2 применительно к задаче обработки облаков точек LIDAR. Учитывая требования горнодобывающей отрасли к надежности, масштабируемости и устойчивости к отказам, рассмотрены ключевые компоненты обеих версий ROS, их архитектурные различия, ограничения ROS1 и преимущества ROS2. Особое внимание уделено реализации конвейера предобработки LiDAR-данных в ROS2, вопросам интеграции с модулями SLAM и обнаружения объектов, а также проблемам конфигурации DDS и настройки QoS. Также рассматриваются сложности, возникающие при переходе с ROS1 на ROS2, и предлагаются рекомендации по их преодолению, включая использование мостов для совместной работы узлов. Представленные результаты могут быть полезны при проектировании надежных и эффективных систем управления беспилотными транспортными средствами в условиях карьеров и других сложных производственных сред.

**Ключевые слова:** ROS, LiDAR, облака точек, SLAM, pointcloud\_preprocessor, RViz.

**DOI:** 10.21293/1818-0442-2025-28-3-132-138

В последние годы беспилотные карьерные самосвалы стали важным звеном в процессе автоматизации горнодобывающих работ. Их внедрение позволяет существенно повысить как производительность, так и уровень безопасности в условиях, где традиционная техника сталкивается с серьезными ограничениями. Основу автономного функционирования таких машин составляет система восприятия окружающей среды, использующая данные, полученные с лидаров. Эти датчики формируют облака точек высокой плотности, что требует оперативной и устойчивой обработки значительных объемов информации в режиме реального времени.

Работа с такими данными осложняется факторами внешней среды – пылью, вибрациями, нестабильным соединением – и требует продуманной архитектуры программного обеспечения.

Долгое время в разработке автономных систем, включая робототехнику и беспилотные транспортные средства, применялась Robot Operating System первого поколения (ROS1). Однако с усложнением задач и ростом требований к отказоустойчивости и масштабируемости возникла необходимость в переходе к более современному решению – ROS2. Эта система предлагает усовершенствованные механизмы обмена сообщениями, поддержку работы в реальном времени и расширенные возможности для построения распределенных решений. В условиях, характерных для работы карьерных самосвалов, данные преимущества особенно важны – в частности, при параллельной обработке информации от нескольких лидаров и других сенсоров.

Тем не менее, несмотря на очевидные плюсы, внедрение ROS2 сопровождается определенными трудностями. Разработчики сталкиваются с проблемами при установке и настройке системы, особенно в распределенных средах с высокой нагрузкой. Кроме того, часть функционала, реализованного в ROS1, может отсутствовать в новой версии или работать нестабильно, что требует доработки и адаптации существующих программных решений.

Цель данной статьи – провести сравнительный анализ архитектур ROS1 и ROS2 в контексте обработки облаков точек лидара для беспилотных карьерных самосвалов. Понимание различий между ROS1 и ROS2 позволит разработчикам и инженерам принимать обоснованные решения при выборе архитектуры для систем автономных карьерных самосвалов, обеспечивая эффективную и надежную работу в сложных условиях горнодобывающей промышленности.

### Обзор различий между ROS1 и ROS2

В первую очередь, рассмотрим архитектуру ROS1 и ROS2. В схеме на рис. 1 наглядно продемонстрированы ключевые различия между этими двумя версиями Robot Operating System, особенно это касается использования DDS (Data Distribution Service).

В левой части схемы видно, как в ROS1 все узлы координируются центральным процессом `roscore` с использованием протоколов TCPROS/UDPROS, тогда как в правой части демонстрируется децентрализованная модель ROS2, построенная на DDS (Data Distribution Service) [1] и абстрагированная слоями клиентских библиотек (`rclcpp`, `rclpy`) и `intra-process-коммуникации` [2]. Переход от единой точки отказа к распределенной системе обмена сообщениями без единого центрального узла закладывает основу для высокой отказоустойчивости и масштабируемости, что особенно важно при передаче больших объемов данных облаков точек лидара в условиях ограниченной пропускной способности и жестких требований к задержкам

ROS1, представляющий собой модульную и расширяемую среду разработки робототехнических приложений, на протяжении многих лет служил эталоном в сообществе исследователей и инженеров. Однако по мере усложнения задач, связанных с обеспечением высокой надежности, горизонтальной масштабируемости и строгих требований к задержкам, выявились ограничения базовой архитектуры ROS1. Именно эти ограничения послужили отправной точ-

кой для появления ROS2, в котором заложены принципы распределенного взаимодействия узлов, встроенной поддержки качественных параметров передачи сообщений и расширенных механизмов реального времени. В условиях обработки высокоплотных облаков точек лидара эти отличия приобретают особую

значимость, поскольку от эффективности обмена данными и устойчивости коммуникации напрямую зависит точность картографирования, надежность локализации и безопасность навигации автономных транспортных платформ в горной среде [3–7].

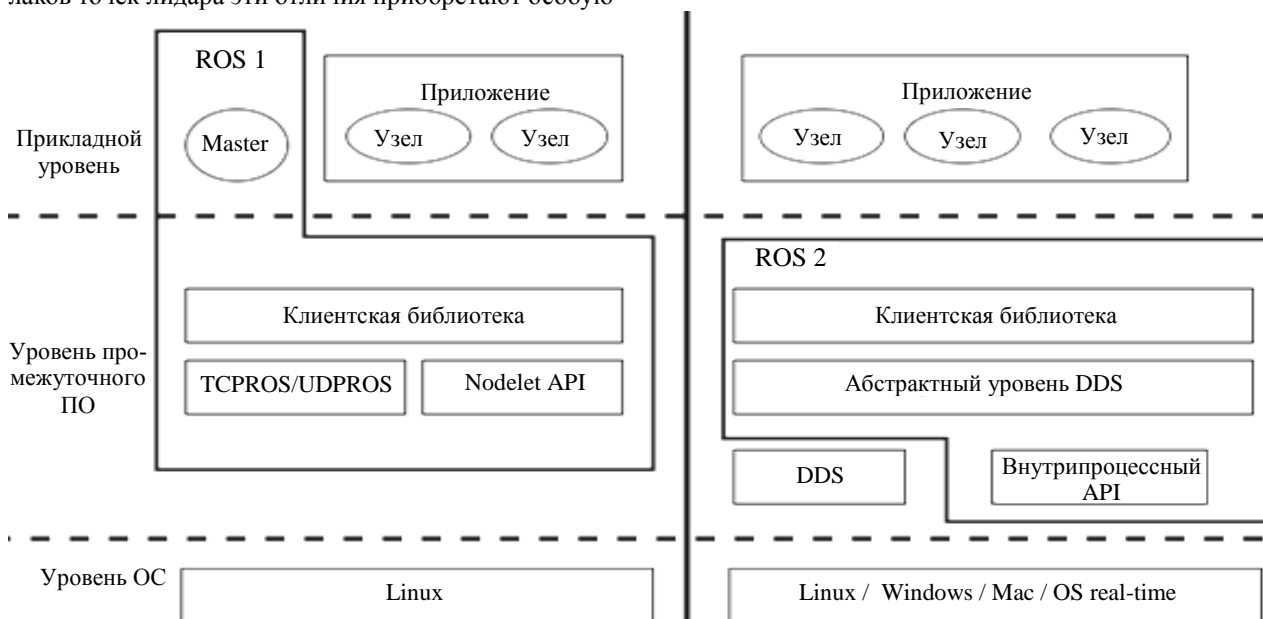


Рис. 1. Архитектура ROS1 и ROS2

### Основные компоненты ROS1

ROS1 построена на централизованной архитектуре, в которой *roscore* выступает в роли единого координационного узла. Между процессами обмен осуществляется по протоколам TCPROS и UDPROS. Для работы с облаками точек лидаров в ROS1 широко применяется пакет *pcl\_gos* – он оборачивает функционал Point Cloud Library (PCL) и предоставляет готовые фильтры, алгоритмы сегментации и другие инструменты обработки данных. Несмотря на свою модульность, ROS1 не обладает встроенными средствами поддержки жестких временных ограничений и масштабируемых распределенных систем, что ограничивает ее применение в задачах с высокими требованиями к надежности и производительности.

#### Обработка облаков точек в ROS1.

Пакет *pcl\_gos* интегрируется с ROS посредством типового сообщения *sensor\_msgs/PointCloud2*, позволяя выполнять предобработку (фильтрацию по пространственным секторам, удаление выбросов, понижение плотности) и последующую сегментацию облаков точек. Для интроспекции и отладки данных в реальном времени используется инструмент RViz, способный визуализировать облака точек и результаты алгоритмов обработки. Однако при работе с облаками высокой плотности или в условиях ограниченных вычислительных мощностей возможно появление задержек в отображении и обработке данных.

#### Ключевые ограничения ROS1:

1. Централизация управления. Зависимость от единого узла *roscore* создает точку отказа и усложняет горизонтальное масштабирование системы.

2. Отсутствие QoS-параметров. ROS1 не позволяет настраивать политику доставки сообщений (надежность, задержки, порядок), что критично при передаче больших объемов данных в реальном времени.

3. Ограничения реального времени. Система не заложена изначально для работы с жесткими временными ограничениями, что может приводить к непредсказуемым задержкам при приеме и обработке облаков точек.

4. Накладные расходы при конвертации форматов. Частые сериализация/десериализация между различными сообщениями и типами данных PCL увеличивает время обработки.

5. Устаревание инструментов. Современные методы фильтрации и классификации облаков точек разрабатываются с прицелом на ROS2 и обычно требуют больших усилий для портирования в ROS1.

6. Сокращение поддержки. С переходом сообщества на ROS2 развитие ROS1 замедляется: исправления и новые функции появляются реже, что создает технический долг в существующих проектах.

В ROS2 полностью отказались от концепции центрального узла (*roscore*), свойственной ROS1, и перешли на стандарт DDS. Такая схема устраняет единую точку отказа и позволяет любому узлу непосредственно находить и связываться с другими участниками системы, что критично для распределенных установок с множественными сенсорными станциями и вычислительными единицами. Использование DDS также открывает доступ к тонкой настройке параметров качества обслуживания (QoS) – надежности

доставки, истории сообщений, глубины очереди и т.д. Это дает возможность задать жесткие гарантии задержки и минимизировать вариативность времени передачи данных от лидара до потребляющих узлов, что важно при работе подвижных платформ в условиях горного рельефа [8].

### Новые технологии в ROS2

*Конвейер предобработки облаков точек.* В экосистеме ROS2 разработан пакет `pointcloud_preprocessor`, входящий в состав `Autoware`, который реализует модульную архитектуру этапов предобработки облаков точек [9]. Общая схема интеграции данных LiDAR, камер и инерциальных датчиков в единую модель окружения часто строится по принципам модульного подхода сенсорного фьюжна в реальном времени. Рекомендуемый конвейер включает:

- фильтрацию по области интереса (`region of interest clipping`);
- коррекцию искажений, вызванных движением лидара (`motion distortion correction`);
- удаление выбросов (`outlier removal`);
- понижение плотности точек (`downsampling`);
- приведение облака к единой системе координат (`coordinate frame transformation`);
- слияние облаков от нескольких сенсоров (`multi-sensor data fusion`).

Такой подход обеспечивает воспроизводимость и повторное использование отдельных модулей при изменении аппаратной конфигурации самосвала или алгоритмических требований [10].

*Интеграция с LiDAR SLAM и объектным обнаружением.*

ROS2 широко применяется совместно с современными LiDAR SLAM-решениями, такими как LOAM и LeGO-LOAM, обеспечивая высокоточную локализацию и построение трехмерных карт в режиме реального времени. Встроенные DDS-механизмы гарантируют своевременную доставку облаков в SLAM-узлы даже при высоких нагрузках [11].

Параллельно с SLAM активно развиваются ROS2-узлы для детекции объектов в облаках точек с применением нейросетевых моделей – в частности, NVIDIA предложила реализацию узла для `PointPillars`, оптимизированного под `TensorRT`. Для обучения и тестирования таких алгоритмов широко используются высокоразрешающие LiDAR-данные, предоставляющие аннотированные 3D-облака точек в условиях карьерного рельефа. Узел принимает сообщения типа `sensor_msgs/PointCloud2`, выполняет инференс и публикует результаты в формате ограничивающих 3D-боксов, что позволяет объединить восприятие и планирование траектории в единую конвейерную цепочку [12].

*Преимущества ROS2 для беспилотных карьерных самосвалов.*

Применение ROS2 в системах автономного вождения, таких как беспилотные карьерные самосвалы, открывает ряд существенных возможностей, востребованных в условиях сложной производственной среды. Среди ключевых преимуществ можно выделить следующие:

1. Масштабируемость: архитектура ROS2 позволяет эффективно работать с большим числом сенсоров и вычислительных узлов, не снижая производительности системы при росте объема данных и сложности задач.

2. Надежность: продвинутые механизмы взаимодействия между компонентами, а также поддержка контроля качества обслуживания (QoS) обеспечивают устойчивую работу системы в условиях высоких механических нагрузок, запяленности и ограниченной сетевой пропускной способности.

3. Гибкость: платформа предоставляет широкие возможности для внедрения и замены алгоритмов обработки информации, что особенно важно при необходимости оперативной адаптации системы под новые задачи или сенсорные модули без серьезных изменений в архитектуре.

4. Совместимость: ROS2 поддерживает взаимодействие с широким спектром аппаратных и программных компонентов – от лидаров и камер до специализированных модулей обработки и управления, что упрощает интеграцию и расширение функциональности.

Несмотря на значительные достоинства, использование ROS2 связано и с рядом сложностей [13].

1. Сложности с DDS и настройкой QoS.

Одной из наиболее заметных является работа с DDS (Data Distribution Service), на которой основана коммуникационная модель ROS2. Хотя DDS обеспечивает необходимую гибкость и масштабируемость, настройка параметров качества обслуживания (QoS) может представлять собой нетривиальную задачу. Различия между реализациями DDS, особенности их конфигурации и взаимодействия нередко требуют глубокого понимания и индивидуального подхода при разработке распределенных систем.

Также имеются проблемы с совместимостью: некоторые пользователи сообщают о трудностях при подключении инструментов визуализации, таких как RViz, к роботам, использующим ROS2, из-за особенностей работы DDS.

2. Ограничения при передаче облаков точек между машинами.

При использовании ROS2 для передачи данных облаков точек между различными машинами могут возникать следующие проблемы:

- потери данных: при передаче больших объемов данных, таких как облака точек, через беспроводные сети возможны потери пакетов, что приводит к неполным или искаженным данным на принимающей стороне [10];

- высокая нагрузка на сеть: без оптимизации и настройки параметров QoS передача больших данных может перегружать сеть, особенно в условиях ограниченной пропускной способности;

3. Ограниченная поддержка некоторых данных в сообщениях `PointCloud2` [14].

В некоторых случаях при использовании ROS2 наблюдаются ограничения в поддержке определенных полей данных в сообщениях `PointCloud2`. Напри-

мер, отсутствие информации об интенсивности: некоторые пользователи сообщают, что при публикации облаков точек в ROS2 отсутствует поле интенсивности, даже если сенсор предоставляет эту информацию [15].

4. Проблемы с производительностью при обработке больших сообщений.

Обработка больших сообщений, таких как облака точек, может вызывать проблемы с производительностью. В частности, необходимость композиции узлов: для эффективной обработки больших сообщений рекомендуется использовать композицию узлов, что может усложнить архитектуру системы.

5. Зависимость от реализации DDS.

Производительность и надежность системы в ROS2 могут сильно зависеть от выбранной реализации DDS: разные реализации DDS могут демонстрировать различные уровни производительности и надежности, что требует тщательного выбора и тестирования.

6. Ограниченная поддержка и документация для некоторых функций.

Некоторые функции и возможности ROS2 могут иметь ограниченную поддержку или недостаточную документацию: отмечено, что для некоторых функций ROS2, особенно новых или специфических, отсутствуют подробные руководства или примеры использования.

Ниже в таблицу сведены основные параметры ROS1 и ROS2 в целях более удобного сравнения.

*Архитектура и коммуникация:* в ROS1 применяется централизованная архитектура с roscore, это может стать причиной «узких мест» в распределенных системах. В отличие от этого ROS2 основан на DDS, обеспечивая децентрализованную коммуникацию между узлами без необходимости в центральном координаторе.

*Поддержка реального времени:* ROS2 предоставляет улучшенную поддержку реального времени благодаря возможности настройки QoS, что особенно важно для систем, обрабатывающих данные от лидаров в реальном времени.

*Обработка облаков точек:* в ROS1 обработка облаков точек осуществляется с использованием pcl\_ros и nodelet, что может вызывать проблемы с производительностью при больших объемах данных. ROS2 предлагает более эффективные и гибкие инструменты, такие как pointcloud\_preprocessor, для обработки облаков точек.

*Формат данных PointCloud2:* ROS2 позволяет настраивать формат точек, выбирая необходимые поля и типы данных, что улучшает совместимость с различными сенсорами и алгоритмами обработки.

*Сжатие и пропускная способность:* в ROS2 требуется дополнительная настройка для достижения эффективности сжатия данных, аналогичной ROS1. Без оптимизации возможна большая нагрузка на сеть.

**Сравнительная таблица: ROS1 vs ROS2 для обработки облаков точек лидара**

Критерий	ROS1	ROS2
Архитектура	Централизованная: использует roscore для координации узлов	Децентрализованная: основана на DDS, обеспечивает прямую коммуникацию между узлами без центрального координатора
Коммуникация	Использует TCPROS/UDPROS; ограниченные возможности настройки QoS	Основана на DDS, предоставляет гибкие политики QoS для настройки надежности, задержек и других параметров коммуникации
Поддержка реального времени	Ограниченная; не предназначена для систем с жесткими требованиями к времени отклика	Улучшенная поддержка реального времени благодаря использованию DDS и возможности настройки QoS
Обработка облаков точек	Использует pcl_ros и nodelet для обработки; возможны проблемы с производительностью при больших объемах данных	Предоставляет пакеты, такие как pointcloud_preprocessor, с улучшенной производительностью и гибкостью обработки [8]
Формат данных PointCloud2	Стандартный формат с полями x, y, z, intensity и ring; возможны проблемы с совместимостью при использовании пользовательских типов точек	Поддерживает настраиваемые форматы точек, включая возможность выбора полей и типов данных, улучшая совместимость и эффективность
Сжатие и пропускная способность	Эффективное сжатие данных; меньшая нагрузка на сеть	Требует дополнительной настройки для достижения аналогичной эффективности сжатия; возможна большая нагрузка на сеть без оптимизации
Интеграция с SLAM и обнаружением объектов	Поддерживает интеграцию с алгоритмами SLAM, такими как Gmapping и Cartographer	Поддерживает современные алгоритмы SLAM, такие как SLAM Toolbox, и интеграцию с нейронными сетями для обнаружения объектов
Поддержка и обновления	Поддержка постепенно прекращается; ограниченные обновления и новые функции	Активная поддержка и развитие; регулярные обновления и новые функции

*Интеграция с SLAM и обнаружением объектов:* ROS2 поддерживает современные алгоритмы SLAM и интеграцию с нейронными сетями для обнаружения объектов, что расширяет возможности системы в контексте автономного вождения.

*Поддержка и обновления:* ROS1 постепенно теряет поддержку, и новые функции разрабатываются в

основном для ROS2, что делает его более перспективным выбором для будущих проектов.

**Мост между ROS1 и ROS2**

Для поэтапного перехода от ROS1 к ROS2 разработан пакет ros1\_bridge, обеспечивающий двунаправленную трансляцию сообщений, сервисов и действий (actions) между двумя версиями фреймворка. Это

позволяет на время миграции запускать в единой системе как узлы ROS1, так и узлы ROS2, обеспечивая совместимость при постепенном переписывании компонентов.

Существует несколько вариантов мостов:

#### 1. Динамический мост (dynamic\_bridge).

При запуске без предварительной генерации описаний типов dynamic\_bridge автоматически обнаруживает все опубликованные в системе типы сообщений, сервисов и действий и создает соответствующие преобразователи «на лету». Такой подход минимизирует ручную конфигурацию и удобен для систем с большим набором стандартных типов, в том числе sensor\_msgs/PointCloud2 для обмена облаками точек лидара и tf2\_msgs/TFMessage для трансляции кадров координат.

Пример: запустив `ros2 run ros1_bridge dynamic_bridge`, пользователь может незамедлительно получать в ROS2-процессах данные с темы `/velodyne_points`, публикуемые ROS1-узлом драйвера лидара, и наоборот.

#### 2. Статический мост (static\_bridge).

Для повышения производительности и сокращения времени старта можно заранее сгенерировать мост только для избранных типов. При этом в коде на C++ создаются конкретные функции-контейнеры для каждого сообщения, сервиса или действия, что исключает накладные расходы на обнаружение типов во время выполнения.

Пример: в проекте автономного карьерного самосвала может быть настроен статический мост для тем обновления карты (`/map_updates`) и сервисов постановки задач навигации (`/navigate_to_pose`), что гарантирует минимальную задержку при обмене большими пакетами данных между ROS1-узлами (ответающими за SLAM) и ROS2-клиентами (управляющими планированием траектории).

#### 3. Мост для действий (actions).

Благодаря поддержке `actionlib` в ROS1 и `rclcpp_action` в ROS2 мост позволяет координировать длительные операции, такие как сбор и агрегация облаков точек с нескольких лидарами. При этом статус выполнения задачи, результат и промежуточные отклики транслируются между системами без потери семантики.

Такие мосты полезны в гибридных робототехнических системах, где накопленные LiDAR-данные продолжают записываться и обрабатываться в среде ROS1, а новые сенсорные модули и алгоритмы управления разворачиваются на платформе ROS2, мосты между версиями обеспечивают сквозную совместимость и непрерывность обмена информацией.

Во-первых, они позволяют без переработки сохранять все наработки по предобработке и фильтрации облаков точек в ROS1, автоматически транслируя сообщения `sensor_msgs/PointCloud2` в ROS2-контекст.

Во-вторых, мосты обеспечивают синхронизацию распределенных подсистем: команды управления и данные позиционирования из ROS2-узлов

поступают к LiDAR-модулю на ROS1 без разрыва временных меток, что минимизирует рассогласование восприятия и управления.

#### Заключение

Таким образом, переход от централизованной архитектуры ROS1 к децентрализованной модели ROS2 на основе DDS открывает новые возможности для построения надежных и масштабируемых систем обработки LiDAR-данных. Вместо одного узла координации теперь узлы обмениваются сообщениями напрямую, что снижает вероятность полного отказа системы и упрощает параллельную работу множества сенсоров и вычислительных модулей.

Вместе с тем внедрение ROS2 налагает свои требования: настройка различных реализаций DDS, тонкая корректировка политик QoS и обеспечение стабильной передачи крупных объемов точечных облаков требуют внимания и опыта. На практике это означает необходимость тщательно продуманного плана миграции, поэтапной проверки компонентов и регулярного мониторинга сетевой и вычислительной нагрузки.

В перспективе дальнейшие усилия будут направлены на оптимизацию обработки LiDAR-данных в ROS2 – от упрощения конвейеров предобработки до совершенствования мостовых решений для плавного взаимодействия с узлами ROS1. Развитие инструментов диагностики и анализа производительности позволит создавать автономные карьерные самосвалы, способные надежно функционировать в самых суровых условиях горнодобывающих предприятий.

Научный вклад данного исследования:

1. Проведен систематический сравнительный анализ архитектур ROS1 и ROS2 с точки зрения их применимости для задач обработки облаков точек LiDAR, выявлены фундаментальные ограничения централизованной модели ROS1 и преимущества децентрализованного обмена данными в ROS2.

2. Разработана концептуальная схема обработки облаков точек в ROS2, учитывающая особенности DDS и механизмы QoS, обеспечивающая повышение надежности и детерминированности вычислений в реальном времени.

3. Предложен подход к интеграции ROS2 с подсистемами локализации и восприятия, демонстрирующий возможность построения масштабируемых и отказоустойчивых киберфизических систем для автономных транспортных средств.

4. Сформулированы теоретические положения и рекомендации, определяющие направления дальнейшей эволюции ROS2 как платформы для распределенной обработки сенсорных данных в экстремальных условиях эксплуатации.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации по соглашению от 30.09.2022 г. № 075-15-2022-1198 с ФГБОУ ВО «Кузбасский государственный технический университет им. Т.Ф. Горбачева» Комплексной научно-технической программы полного инновационного цикла «Разработка

и внедрение комплекса технологий в областях разведки и добычи твердых полезных ископаемых, обеспечения промышленной безопасности, биоремедиации, создания новых продуктов глубокой переработки из угольного сырья при последовательном снижении экологической нагрузки на окружающую среду и рисков для жизни населения» (КНТП «Чистый уголь – Зеленый Кузбасс»), утвержденной Распоряжением Правительства Российской Федерации от 11 мая 2022 г. № 1144-р в рамках реализации мероприятия «Разработка и создание беспилотного карьерного самосвала челночного типа грузоподъемностью 220 тонн» в части выполнения научно-исследовательских и опытно-конструкторских работ.

#### Литература

1. G330 ROS-1 Tutorial: Point Cloud [Электронный ресурс]: Orbbec. – Режим доступа: <https://www.orbbec.com/docs/g330-ros-1-tutorial-point-cloud/> (дата обращения: 22.05.2025).
2. Point Cloud Pre-processing Design [Электронный ресурс]: Autoware Foundation – Autoware Documentation. – Режим доступа: <https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture/sensing/data-types/point-cloud/> (дата обращения: 22.05.2025).
3. Дубинкин Д.М. Создание алгоритма обработки данных систем беспилотного управления карьерного самосвала для построения высокоточных 3D-карт местности / Д.М. Дубинкин, В.Ю. Садовец, И.С. Сыркин, Е.В. Коршунова // Уголь. – 2024. – № S11(1187). – С. 116–122. DOI: 10.18796/0041-5790-2024-11S-116-122.
4. Developing environmentally adapted simulators for autonomous mining dump trucks: a Multi-criteria approach to enhance sustainability and ecological safety / I. Syrkin, V. Sadovets, E. Korshunova et al. // International Journal of Ecosystems and Ecology Science. – 2024. – Vol. 14, No. 4. – P. 181–190. DOI: 10.31407/ijees14.422.
5. Выбор архитектуры системы управления беспилотным транспортным средством для движения по закрытым территориям / И.В. Лобачев, М.А. Рыженко, Г.А. Арутюнян, А.С. Муравьев, Д.М. Дубинкин // Вестник Казан. гос. техн. ун-та им. А.Н. Туполева. – 2024. – Т. 80, № 2. – С. 87–91.
6. Оптимизация показателей работы роботизированного экскаваторно-автомобильного комплекса разреза / Ю.Е. Воронов, А.Ю. Воронов, Д.М. Дубинкин, О.С. Максимова // Уголь. – 2024. – № 7(1182). – С. 62–67. DOI: 10.18796/0041-5790-2024-7-62-67.
7. Анализ показаний телеметрии при эксплуатации карьерных самосвалов грузоподъемностью 220 тонн / А.И. Бокарев, В.А. Дианов, А.Б. Карташов и др. // Горное оборудование и электромеханика. – 2024. – № 5(175). – С. 52–60. DOI: 10.26730/1816-4528-2024-5-52-60.
8. ROS2 with multiple machines doesn't work properly with point clouds [Электронный ресурс]: arusso, gvdhoorn // ROS Answers Archive. – 2023. – URL: <https://answers.ros.org/question/416682> (дата обращения: 13.05.2025).
9. Ros2 publish point cloud excluding intensity information [Электронный ресурс]: Vulcan.Shao // NVIDIA Developer Forums. – 04.07.2024. – URL: <https://forums.developer.nvidia.com/t/ros2-publish-point-cloud-excluding-intensity-information/298606> (дата обращения: 11.05.2025).
10. Macenski S. Impact of ROS 2 Node Composition in Robotic Systems / S. Macenski, A. Soragna, M. Carroll, Z. Ge [Электронный ресурс]: arXiv preprint arXiv:2305.09933. – 17.05.2023. – URL: <https://arxiv.org/pdf/2305.09933> (дата обращения: 22.05.2025).
11. Qiu Z. Exploring the performance of ROS 2: A survey of ROS 2 research, libraries and applications / Z. Qiu, H. Wang, M. Li [Электронный ресурс]: arXiv:2411.11607v2 [Preprint]. – 2024. – URL: <https://arxiv.org/html/2411.11607v2> (дата обращения: 22.05.2025).
12. Mortensen A. Robot Operating System: How to model point cloud data in ROS 2 [Электронный ресурс]: Dev.to. – 2023. – URL: <https://dev.to/admantium/robot-operating-system-how-to-model-point-cloud-data-in-ros2-ai> (дата обращения: 18.05.2025).
13. Kin C. ros2\_pointcloud\_processor [Электронный ресурс]: GitHub. – 2024. – URL: [https://github.com/ChristopherKin/ros2\\_pointcloud\\_processor](https://github.com/ChristopherKin/ros2_pointcloud_processor) (дата обращения: 21.05.2025).
14. ROS on DDS [Электронный ресурс]: ROS 2 Design Working Group. – 2023. – URL: [https://design.ros2.org/articles/ros\\_on\\_dds.html](https://design.ros2.org/articles/ros_on_dds.html) (дата обращения: 21.05.2025).
15. Addison J. ROS 2 Architecture Overview [Электронный ресурс]: Automatic Addison. – 2023. – URL: <https://automaticaddison.com/ros-2-architecture-overview/> (дата обращения: 22.05.2025).

---

#### Коршунова Елизавета Вячеславовна

Мл. науч. сотр. научного центра цифровых технологий Кузбасского гос. техн. ун-та им. Т.Ф. Горбачева (КузГТУ) Весенняя ул., 28, г. Кемерово, Россия, 650000  
Тел.: +7-909-513-15-07  
Эл. почта: korshunovae@kuzstu.ru

#### Сыркин Илья Сергеевич

Канд. техн. наук, доцент каф. информационных и автоматизированных производственных систем КузГТУ Весенняя ул., 28, г. Кемерово, Россия, 650000  
Тел.: +7-903-909-24-91  
Эл. почта: syrkin@kuzstu.ru

#### Садовец Владимир Юрьевич

Канд. техн. наук, доцент каф. информационных и автоматизированных производственных систем КузГТУ Весенняя ул., 28, г. Кемерово, Россия, 650000  
ORCID: 0000-0001-6056-9504  
Тел.: +7-950-263-84-75  
Эл. почта: svyu.pmh@kuzstu.ru

Поступила в редакцию: 03.07.2025.

Принята к публикации: 10.11.2025.

#### Korshunova E.V., Syrkin I.S., Sadovets V.Yu. Architectural Comparison of ROS1 and ROS2 in the Context of Processing LiDAR Point Cloud Data

This paper provides a comparative analysis of the architectural features of ROS1 and ROS2 as applied to the task of processing LiDAR point clouds. Considering the mining industry's requirements for reliability, scalability, and fault tolerance, the key components of both ROS versions, their architectural differences, the limitations of ROS1, and the advantages of ROS2 are discussed. Special attention is given to the implementation of the LiDAR data preprocessing pipeline in ROS2, integration with SLAM and object detection modules, as well as DDS configuration and QoS settings. The article also discusses difficul-

ties encountered when migrating from ROS1 to ROS2 and offers recommendations for overcoming them, including the use of bridges for joint operation of nodes. The presented results can be useful in designing reliable and efficient control systems for unmanned vehicles operating in open-pit mines and other complex industrial environments.

**Keywords:** ROS, LiDAR, point cloud, SLAM, pointcloud\_preprocessor, RViz.

**DOI:** 10.21293/1818-0442-2025-28-3-132-138

### References

1. G330 ROS-1 Tutorial: Point Cloud [Электронный ресурс]: Orbbec. – Available at: <https://www.orbbec.com/docs/g330-ros-1-tutorial-point-cloud/> (Accessed: 22 May 2025).

2. Point Cloud Pre-processing Design [Электронный ресурс]: Autoware Foundation – Autoware Documentation. Available at: <https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture/sensing/data-types/point-cloud/> (Accessed: 22 May 2025).

3. Dubinkin D.M., Sadovets V.Yu., Syrkin I.S., Korshunova E.V. [Development of an algorithm for processing data from autonomous control systems of mining dump trucks to construct high-precision 3D-terrain maps] *Russian Coal Journal Ugol'*, 2024, no. S11 (1187), pp. 116–122. DOI: 10.18796/0041-5790-2024-11S-116-122.

4. Syrkin I., Sadovets V., Korshunova E. et al. Developing environmentally adapted simulators for autonomous mining dump trucks: a multi-criteria approach to enhance sustainability and ecological safety. *International Journal of Ecosystems and Ecology Science*, 2024, vol. 14, no. 4, pp. 181–190. DOI: 10.31407/ijees14.422.

5. Lobachev I.V., Ryzhenko M.A., Harutyunyan G.A., Muraviev A.S., Dubinkin D.M. *Vybor arhitektury sistemy upravleniya bespilotnym transportnym sredstvom dlya dvizheniya po zakrytym territoriyam* [The choice of the architecture of the control system of a gravity-free vehicle for driving in closed territories]. *Vestnik KGTU im. A.N. Tupoleva*, 2024, vol. 80, no. 2, pp. 87–91.

6. Voronov Yu.E., Voronov A.Yu., Dubinkin D.M., Maksimova O.S. *Optimizatsiya pokazatelei raboty robotizirovannogo ekskavatorno-avtomobil'nogo kompleksa razreza* [Optimization of performance indicators of the robotic excavator and automobile complex of the mine] *Russian Coal Journal Ugol'*, 2024, no. 7 (1182), pp. 62–67. DOI 10.18796/0041-5790-2024-7-62-67.

7. Bokarev A.I., Dianov V.A., Kartashov A.B., Harutyunyan G.A., Dubinkin D.M., Pashkov D.A. [Analysis of telemetry readings during operation of 220-ton dump trucks]. *Mining Equipment and Electromechanics*, 2024, no. 5 (175), pp. 52–60. DOI 10.26730/1816-4528-2024-5-52-60.

8. ROS2 with multiple machines doesn't work properly with point clouds / arusso, gvdhoorn. ROS Answers Archive, 2023. Available at: <https://answers.ros.org/question/416682> (Accessed: 13 May 2025).

9. Ros2 publish point cloud excluding intensity information. Vulcan.Shao. NVIDIA Developer Forums, 2024. Available at: <https://forums.developer.nvidia.com/t/ros2-publish-point-cloud-excluding-intensity-information/298606> (Accessed: 11 May 2025).

10. Macenski S., Soragna A., Carroll M., Ge Z. Impact of ROS 2 Node Composition in Robotic. *arXiv preprint arXiv:2305.09933*, 2023. Available at: <https://arxiv.org/pdf/2305.09933> (Accessed: 22 May 2025).

11. Qiu Z., Wang H., Li M. Exploring the performance of ROS 2: A survey of ROS 2 research, libraries and applications. *arXiv:2411.11607v2*, 2024. Available at: <https://arxiv.org/html/2411.11607v2> (Accessed: 22 May 2025).

12. Mortensen A. Robot Operating System: How to model point cloud data in ROS 2. *Dev.to*, 2023. Available at: <https://dev.to/admantium/robot-operating-system-how-to-model-point-cloud-data-in-ros2-ai> (Accessed: 18 May 2025).

13. Kin C. ros2\_pointcloud\_processor. *GitHub*, 2024. Available at: [https://github.com/ChristophKin/ros2\\_pointcloud\\_processor](https://github.com/ChristophKin/ros2_pointcloud_processor) (Accessed: 21.05.2025).

14. ROS on DDS. ROS 2 Design Working Group, 2023. Available at: [https://design.ros2.org/articles/ros\\_on\\_dds.html](https://design.ros2.org/articles/ros_on_dds.html) (Accessed: 21 May 2025).

15. Addison J. ROS 2 Architecture Overview. *Automatic Addison*, 2023. Available at: <https://automaticaddison.com/ros-2-architecture-overview/> (Accessed: 22 May 2025).

---

### Elizaveta V. Korshunova

Junior research fellow at the Scientific Center for Digital Technologies of the Kuzbass State Technical University named after T.F. Gorbachev (KuzSTU)  
Vesenyaya st., 28, Kemerovo, Russia, 650000  
Phone: +7-909-513-15-07  
Email: korshunovaev@kuzstu.ru

### Ilya S. Syrkin

Associate Professor, Ph.D. in Engineering, KuzSTU  
Vesenyaya st., 28, Kemerovo, Russia, 650000  
Phone: +7-903-909-24-91  
Email: syrkin@kuzstu.ru

### Vladimir Yu. Sadovets

Associate Professor, Ph.D. in Engineering, KuzSTU  
Vesenyaya st., 28, Kemerovo, Russia, 650000  
ORCID: 0000-0001-6056-9504  
Phone: +7-950-263-84-75  
Email: svyu.pmh@kuzstu.ru

Received: 03.07.2025.

Accepted: 10.11.2025.