

УДК 004.021:004.75

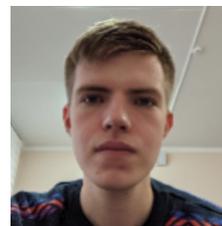
ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ BIG DATA ПРИ ПОСТРОЕНИИ СИСТЕМЫ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ



М.А. Гергенсон
Студент факультета
компьютерных систем и
сетей БГУИР
maratgerghenson@gmail.com



С.Н. Нестеренков
Декан факультета
компьютерных систем
и сетей БГУИР,
кандидат технических
наук, доцент
s.nesterenkov@bsuir.by



И.С. Тарасюк
Ассистент кафедры
электронных
вычислительных машин
БГУИР
i.tarasiuk@bsuir.by

М.А. Гергенсон

Студент четвертого года обучения по специальности "Вычислительные машины, системы и сети" БГУИР.

С.Н. Нестеренков

Кандидат технических наук, доцент, декан факультета компьютерных систем и сетей Белорусского государственного университета информатики и радиоэлектроники, доцента кафедры программного обеспечения информационных технологий. Автор публикаций на тему машинного обучения, алгоритмов принятия решений, искусственных нейронных сетей и автоматизации.

И.С. Тарасюк

Окончил БГУИР в 2021 году по специальности "Вычислительные машины, системы и сети", магистрант первого года обучения по специальности "Системы и сети инфокоммуникаций" БГУИР.

Аннотация. В ходе разработки беспилотного мобильного робота возникла проблема представления и анализа большого потока неструктурированных данных, собранными подсистемами устройств. Для ее решения была создана система обработки данных, позволяющее развивать дальше разработанное устройство при помощи технологий анализа больших данных.

Ключевые слова: мобильный робот, подсистема, ROS, Hadoop, Spark.

Введение.

Мобильный робот — это робот, способный передвигаться. Мобильная робототехника обычно считается подполем робототехники и информатики.

Мобильные роботы могут перемещаться в своей среде и не привязаны к одному физическому местоположению. Мобильные роботы могут быть «автономными» (AMR — автономный мобильный робот), что означает, что они способны осуществлять навигацию по неконтролируемой среде без необходимости использования физических или электромеханических устройств управления. В качестве альтернативы мобильные роботы могут полагаться на устройства управления, которые позволяют им перемещаться по заранее определенному маршруту навигации в относительно контролируемом пространстве (AGV — автономный управляемый автомобиль). Напротив, промышленные роботы обычно более или менее неподвижны, состоящие из сочлененного рычага (многосвязного манипулятора) и захватного узла (или конечного эффектора), прикрепленного к неподвижной поверхности.

Мобильные роботы стали более распространенными в коммерческих и промышленных условиях. Больницы используют автономные мобильные роботы для перемещения материалов в течение многих лет. В складах установлены мобильные роботизированные системы для эффективного перемещения материалов с полки для хранения в зоны выполнения заказов. Мобильные роботы также являются основным направлением текущих исследований, и почти у каждого крупного университета есть одна или несколько лабораторий, которые ориентированы на исследования мобильных роботов. Мобильные роботы также находятся в промышленных, военных и охранных системах. Внутренние роботы — это потребительские товары, в том числе развлекательные роботы и те, которые выполняют определенные домашние задачи, такие как пылесос или садоводство.

Структура мобильного робота.

Беспилотные мобильные роботы способны передвигаться самостоятельно, благодаря специальному программному обеспечению (ПО) и сенсорам. ПО управляет работой всех систем робота: поворачиванием руля, сменой передач, газом и тормозом. Датчики (сенсоры) собирают информацию об окружающей обстановке, которая ложится в основу действий мобильного робота. Компоненты мобильного робота — это контроллер, программное обеспечение для управления, датчики и исполнительные механизмы. Контроллер обычно представляет собой микропроцессор, встроенный микроконтроллер или персональный компьютер (ПК). Программное обеспечение для мобильного управления может быть либо языком уровня сборки, либо языками высокого уровня, такими как C, C++, Pascal, Fortran или специальным программным обеспечением реального времени. Программное обеспечение беспилотного робота может включать компьютерное зрение и нейросети.

Большое число детекторов наделяет беспилотные роботы сверхчеловеческими способностями. Система, обеспечивающая беспилотное управление робота, включает в себя набор сенсоров, которые обеспечивают зрением машину, видеокамеры, радары, бортовое вычислительное устройство, которое обрабатывает сигнал, получаемый с этих датчиков. На этом устройстве формируется дорожная сцена, то есть устанавливаются объекты, которые находятся вокруг машины. Получается некоторая картинка, в центре которой находится робот, вокруг него находятся разного рода объекты: дома, другие автомобили, дорожные знаки, пешеходы, разметка, то есть всё, что нужно знать для того, чтобы мобильный робот мог в дальнейшем ориентироваться в пространстве: ехать прямо, перестроиться, притормозить, предпринять какие-либо действия, чтобы человека довести из точки А в точку Б, соблюдая все правила дорожного движения и, при этом, не попав в ДТП. Если один из детекторов обнаружил, что есть препятствия, дальше начинается оценка степени опасности этого препятствия, которое находится на траектории движения и оценивается время до столкновения. В полуавтоматическом режиме компьютер оповещает водителя, что впереди опасность и надо притормозить, если водитель не реагирует, то машина начинает тормозить самостоятельно. Обычно устанавливаемые датчики: лидары — дальномер оптического распознавания, радары, камеры, система глобального позиционирования (GPS, Глонасс), Датчики одометрии, гиростабилизатор.

Фреймворк Robot Operating System.

Создание мобильного робота является командной работой, где параллельно ведется разработка нескольких систем, одной из которой и является автопилот, из чего произрастает вопрос о том, как обеспечить слаженную работу разрабатываемых систем. Для этого было решено использовать фреймворк ROS. Robot Operating System (ROS) — это гибкая платформа (фреймворк) для разработки программного обеспечения роботов. Это набор разнообразных инструментов, библиотек и определенных правил, целью которых является упрощение задач разработки ПО роботов. ROS была создана, чтобы

стимулировать совместную разработку программного обеспечения робототехники. Каждая отдельная команда может работать над одной конкретной задачей, но использование единой платформы, позволяет всему сообществу получить и использовать результат работы этой команды для своих проектов. Критериями, благодаря которым командой был выбран именно этот фреймворк, являются активность сообщества, наличие различных библиотек, расширяемость и простота использования. По этим критериям в данный момент равных ROS нет.

Все подсистемы в мобильном роботе связаны посредством ROS. Каждая из подсистем постоянно нуждается в информации, находящейся в другой, например, автопилот нуждается в данных с датчиков, модулей видеонаблюдения, геолокации и позиционирования, при этом всем все эти данные напрямую влияют на состояния автопилота и его поведение. Вся собранная информация о состоянии робота необходима для отслеживания системы, анализа работоспособности в не рассматриваемых на первоначальных этапах разработке ситуаций и сравнение результатов различных версий программы. Таким образом все накопленное придется хранить. ROS позволяет при помощи команды `rosv bag record` перехватывать все сообщения между узлами в указанном месте, однако в итоге мы получаем огромный пласт неструктурированных данных. Отладка такой системы и нахождение закономерностей – определенно тяжелая задача для выполнения вручную, да и данные продолжают неистово накапливаться во время использования.

Преобразование данных ROS в пригодные для анализа данные.

Предполагалось, что файлы пакета ROS должны быть преобразованы в более подходящий формат, прежде чем их можно будет обрабатывать параллельно с помощью таких инструментов, как Hadoop или Spark. Получается, что формат достаточно хорош для обработки с распределенной файловой системой типа HDFS. Для этого предназначен Hadoop `RosbagInputFormat` — разделяемый Hadoop `InputFormat` с открытым исходным кодом для формата файла `rosv bag`.

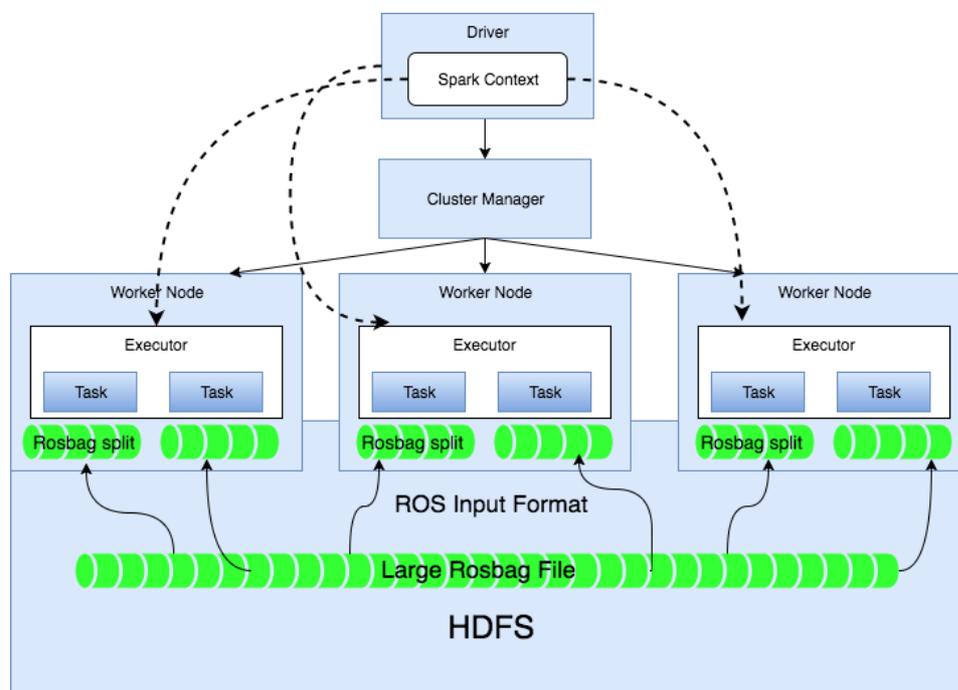


Рисунок 1. Пример разбиения файла Rosbag с Hadoop

Hadoop – это свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Эта основополагающая технология хранения и обработки больших данных (Big Data) является проектом верхнего уровня фонда Apache Software Foundation. Изначально проект разработан на Java в рамках вычислительной парадигмы MapReduce, когда приложение разделяется на большое количество одинаковых элементарных заданий, которые выполняются на распределенных компьютерах (узлах) кластера и сводятся в единый результат.

Классический MapReduce, Apache компонент Hadoop для обработки данных, проводит вычисления в два этапа:

1 Map, когда главный узел кластера распределяет задачи по рабочим узлам.

2 Reduce, когда данные сворачиваются и передаются обратно на главный узел, формируя окончательный результат вычислений.

Пока все процессы этапа Map не закончатся, процессы Reduce не начнутся. При этом все операции проходят по циклу чтение-запись с жесткого диска. Это обуславливает задержки в обработке информации. Таким образом, технология MapReduce хорошо подходит для задач распределенных вычислений в пакетном режиме, но из-за задержек (latency) не может использоваться для потоковой обработки в режиме реального времени. Для решения этой проблемы был создан Apache Spark и другие Big Data фреймворки распределенной потоковой обработки (Storm, Samza, Flink).

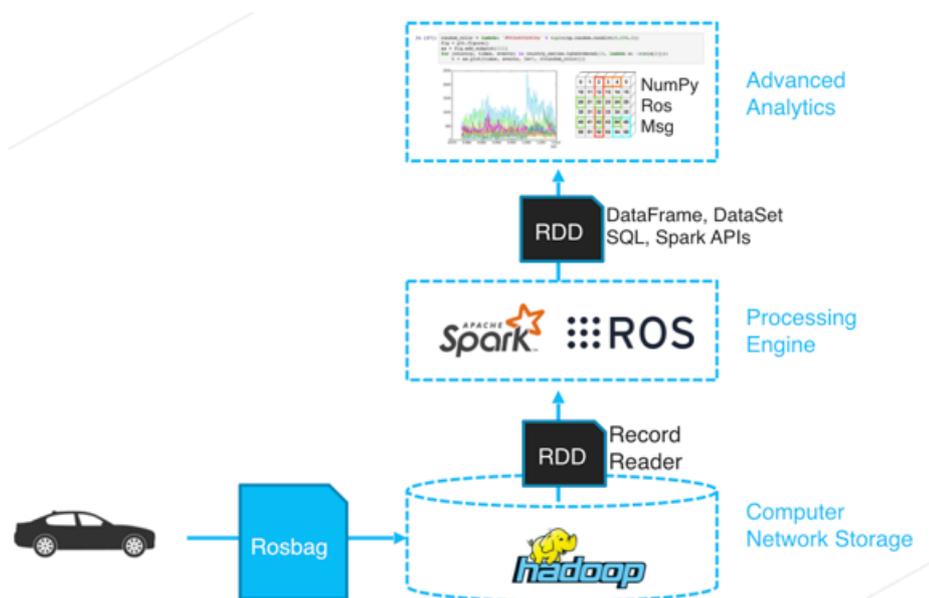


Рисунок 2. Итоговая структура обработки данных мобильного робота

В отличие от классического обработчика ядра Apache Hadoop с двухуровневой концепцией MapReduce на базе дискового хранилища, Spark использует специализированные примитивы для рекуррентной обработки в оперативной памяти. Благодаря этому многие вычислительные задачи реализуются в Spark значительно быстрее. Например, возможность многократного доступа к загруженным в память пользовательским данным позволяет эффективно работать с алгоритмами машинного обучения (Machine Learning).

Благодаря наличию разнопрофильных инструментов для аналитической обработки данных «на лету» (SQL, Streaming, MLLib, GraphX), Spark активно используется в системах

интернета вещей (Internet of Things, IoT) на стороне IoT-платформ, а также в различных бизнес-приложениях, в т.ч. на базе методов Machine Learning.

Заключение.

Возникшую при разработке беспилотного мобильного робота проблему представления и анализа данных работы подсистем машин, препятствующую дальнейшему развитию системы, удалось решить при помощи методов анализа больших данных на базе технологий Apache Hadoop и Spark.

В итоге была получена структура, позволяющая на ее базе реализовывать всевозможные виды систем обработки информации и управления, например, системы автопилотирования на базе методов машинного обучения, представления информации оператору и т.д.

Список литературы

- [1] Марц, Натан Большие данные. Принципы и практика построения масштабируемых систем обработки данных в реальном времени: моногр. / Натан Марц, Джеймс Уоррен. – М.: Вильямс, 2016. – 368 с.
- [2] Фуругян, Меран Алгоритмы планирования вычислений и синтеза систем реального времени / Меран Фуругян. – М.: LAP Lambert Academic Publishing, 2012. – 232 с.
- [3] Хетагуров, Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ). Учебник / Я.А. Хетагуров. – М.: Бином. Лаборатория знаний, 2015. – 240 с.
- [4] Lentin Joseph. Robot Operating System (ROS) for Absolute Beginners: Robotics Programming Made Easy / Lentin Joseph. — Apress, 2018.
- [5] Изучаем Spark. Молниеносный анализ данных / Х. Карау [и др.]. — ДМК Пресс, 2015. — 304 с.
- [6] Беляк, А. А. Анализ производительности технологии Hadoop / А. А. Беляк, С. Н. Нестеренков // BIG DATA and Advanced Analytics = BI DATA и анализ высокого уровня: сб. научных статей VII Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 19-20 мая 2021 года): / редкол.: В. А. Богуш [и др.]. – Минск: Бестпринт, 2021. – С. 343–346.
- [7] Одиноченко, М. И. Использование облачных сервисов для решения задач, связанных с применением технологии BIG DATA / М. И. Одиноченко, С. Н. Нестеренков // BIG DATA and Advanced Analytics = BI DATA и анализ высокого уровня: сб. научных статей VIII Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 11-12 мая 2022 года): / редкол.: В. А. Богуш [и др.]. – Минск: Бестпринт, 2022. С. 252–256.

QUALITY EVALUATION OF INFORMATION TRANSFER IN A DISPATCHING SYSTEM BASED ON MQTT ARCHITECTURE

M.A. Herhenson

Student of the Faculty of Computer systems and networks of BSUIR

S.N. Nesterenkov

Dean of the Faculty of Computer systems and networks of BSUIR, PhD of Technical Sciences, Associate Professor

I.S. Tarasiuk

Assistant of the Department of Electronic Computers of BSUIR

Department of Electronic Computers

Faculty of Computer systems and networks

Belarusian State University of computer science and Radio Electronics, Republic of Belarus

E-mail: maratgergenson@gmail.com

Abstract. During the development of an unmanned mobile robot, the problem of presenting and analyzing a large flow of unstructured data collected by device subsystems arose. To solve it, a data processing system was created that allows further development of the developed device using big data analysis technologies.

Keywords: mobile robot, subsystem, ROS, Hadoop, Spark.