

Заключение. Предлагаемый метод имеет несколько важных аспектов.

- во-первых, показания вибрации получены в реальных условиях эксплуатации, что означает применение стандартных систем мониторинга вибрации. Хотя, с одной стороны, этот метод имеет недостаток в неточности считывания, с другой стороны, он позволяет оператору самостоятельно собирать сигналы без посторонней помощи.

- во-вторых, это связано с основным аспектом метода Маркова, который позволяет оператору выполнять диагностику направляющих подшипников без необходимости приостановки работы направляющего устройства для проведения проверок.

Следовательно, предлагаемый метод имеет существенные преимущества за счет снижения риска отказа и затрат на техническое обслуживание.

Литература

1. Рабинер Л. Р. и Хуанг Б.-Х. Введение в скрытые марковские модели // *IEEE ASSP*. – том 3
2. Ван Г. Ф., Фенг Х. Л. и Лю С. Классификация неисправностей подшипников на основе условного случайного поля // *Удар и вибрация*. – том 20
3. Пэн З., Чу Ф. и Хе Ю. Анализ сигналов вибрации и извлечение признаков на основе переназначенной вейвлет-скалограммы // *Журнал звука и вибрации*. – том 253.
4. Патент 2239803 Российская Федерация, МПК G01L 1/22. Измерение механических колебаний или ультразвуковых, звуковых или инфразвуковых колебаний с использованием средств, чувствительных к излучению, например оптических средств / Дедученко Ф.М., Дылюк А.Г.,

Коновалов И.Л., Липко А.Н.; Заявитель и патентообладатель ОАО «НПО Промавтоматика»; Заявл. 05.06.2002; Опубл. 10.11.2004.

5. Мупшинская А. Вибрационная диагностика неисправностей вращающихся машин
6. Чжан З., Ван Ю. и Ван К. Диагностика неисправностей и прогноз с использованием разложения вейвлет-пакетов, преобразования Фурье и искусственной нейронной сети // *Журнал интеллектуального производства*. – том 24

References

1. L. R. Rabiner i B.-H. Huang, “Vvedenie v skrytye markovskie modeli”, zhurnal IEEE ASSP, tom 3
2. G. F. Van, H. L. Feng i S. Lyu, “Klassifikaciya neispravnostej podshipnikov na osnove uslovnogo sluchajnogo polya”, “Udar i vibraciya”, tom 20
3. Z. Pen, F. CHu i YU. He, “Analiz signalov vibracii i izvlechenie priznakov na osnove perenaznachennoj vejvlet-skalogrammy”, ZHurnal zvuka i vibracii, tom 253
4. Patent 2239803 Rossijskaya Federaciya, MPK G01L 1/22. Izmerenie mekhanicheskikh kolebanij ili ul'trazvukovyh, zvukovyh ili infrazvukovyh kolebanij s ispol'zovaniem sredstv, chuvstvitel'nyh k izlucheniyu, naprimer opticheskikh sredstv / Deduchenko F.M., Dilyuk A.G., Konovalov I.L., Lipko A.N.; Zayavitel' i patentoobladatel' ОАО «NPO Promavtomatika»; Zayavl. 05.06.2002; Opubl. 10.11.2004.
5. A. Mushinskaya, “Vibracionnaya diagnostika neispravnostej vrashchayushchihsy mashin”
6. Z. CHzhan, YU. Van i K. Van, “Diagnostika neispravnostej i prognoz s ispol'zovaniem razlozheniya vejvlet-paketov, preobrazovaniya Fur'e i iskusstvennoj nejronnoj seti”, ZHurnal intelektual'nogo proizvodstva, tom 24

УДК 004.451.42

DOI: 10.34046/aumsuomt105/41

АНАЛИЗ ИНСТРУМЕНТОВ УПРАВЛЕНИЯ И АДМИНИСТРИРОВАНИЯ ОПЕРАЦИОННЫХ СИСТЕМ СЕМЕЙСТВА LINUX И WINDOWS

М.А. Аль-Ханани, кандидат технических наук

И.Д. Соколова, магистрант

А.А. Пучков, аспирант

В данной научной статье выполняется исследование рынка современных инструментов управления и администрирования ОС семейства Linux и Windows, производится сравнительный анализ самых популярных и востребованных из них. Формулируются основные требования к инструментам управления и администрирования, выделяются их основные особенности.

На сегодняшний день существует множество инструментов управления и администрирования ОС. Каждый из них имеет свои преимущества и недостатки. Поэтому выбрать подходящий инструмент иногда бывает очень затруднительно.

Ключевые слова: администрирование, операционная система, сервер.

ANALYSIS OF MANAGEMENT AND ADMINISTRATION TOOLS LINUX AND WINDOWS OPERATING SYSTEMS

M.A. Al-Khanani, I.D.Sokolova, A.A. Puchkov

In this scientific article, the market research of modern management and administration tools of the Linux and Windows OS family is carried out, a comparative analysis of the most popular and demanded of them is carried out. The basic requirements for management and administration tools are formulated, their main features are highlighted.

To date, there are many OS management and administration tools. Each of them has its advantages and disadvantages. Therefore, it is sometimes very difficult to choose the right tool.

Keywords: administration, operating system, server.

Введение

Операционную систему (ОС) можно назвать программным продолжением устройства управления компьютера. ОС скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера. Наиболее популярными ОС являются Linux и Windows.

Каждый из рассматриваемых в статье инструментов предназначается для управления конфигурацией ОС внутри одной сети. Администратор, который находится за сервером, получает возможность отправлять на периферийные машины данные конфигураций. На устройствах обязательно должна быть установлена клиентская часть. Система осуществляет свою деятельность автоматизировано, конфигурируя, полагаясь на те настройки, которые доставил сервер.

Эти программные разработки применяются для администрирования нескольких VPS, управления парком серверов, настройки их серверов с нуля, подключения дополнительных НОД в кластер и прочих важных задач, возложенных на плечи сетевого админа.

В данной статье рассмотрены такие утилиты, как Puppet, CHEF, Ansible и Salt.

Puppet- открытое приложение Puppet позволяет конфигурировать операционные системы для развертывания, управления и обновления компьютерных сетей на тысячах удаленных машин. Решение построено на клиент-серверной архитектуре и заслуженно считается одним из самых функциональных инструментов в своём классе. Используют систему такие именитые корпорации, как PayPal, Google, Reddit, Dell, Oracle и Стэнфордский университет.

Puppet рассчитан на использование в достаточно крупных компаниях, где управление ИТ-инфраструктурой при помощи самостоятельно написанных сценариев может быть затруднено. Приложение оперирует конфигурациями, для составления которых применяется специальный предметно-ориентированный язык.

При помощи Puppet системный администратор может эффективно решать следующие задачи:

- определение уникальных настроек для каждого хоста в сети;
- осуществление непрерывного мониторинга сети;
- управление пользователями;
- настройка приложений.

Плюсы Puppet:

– Автоматизация. Puppet позволяет автоматически конфигурировать операционные системы на удаленных машинах, что избавляет системного инженера десятки раз вручную одинаковым образом настраивать машины, выезжая на места.

– Скорость развертывания. Автоматизация передачи и применения настроек экономит ресурсы пользователя и ускоряет процесс развертывания и обновления сети в десятки раз.

– Кроссплатформенность. Puppet позволяет включать в общую сеть компьютеры под различными операционными системами, и обеспечить их эффективное взаимодействие.

– Поддержка. Puppet имеет техническую поддержку, специалисты которой фиксируют обращения пользователей и работают над исправлением проблем.

– Общая безопасность системы. Puppet обновляет конфигурацию множества узлов системы параллельно, не составляет особых сложностей применить в конфигурировании клиентских операционных систем гибкие настройки безопасности.

Минусы Puppet:

– Квалификация администратора. Если при составлении файла конфигурации допустить ошибку, и затем не проверить этот файл должным образом, можно одним махом "положить" несколько сотен серверов.

– Несовместимость с некоторыми версиями ОС (например, ранние версии Windows NT)

– Уязвимость сервера. Если злоумышленник получит доступ к серверу Puppet, то может сконфигурировать клиентские операционные системы так, как ему заблагорассудится.

CHEF — это система конфигурирования сети, которая основана на файлах конфигурации, которые администратор объединяет в сценарии

поведения сети. Эти сценарии помещаются в хранилище, откуда актуальный набор конфигураций извлекается и устанавливается на клиентские машины в автоматическом режиме. Все операции исполняются с помощью консольного инструмента. Серверная часть решения написана на Erlang, а клиентская — на Ruby.

Программа применяется для первичной настройки и поддержки нормальной работы серверов компании, число которых может составлять десятки или даже сотни.

Особенность CHEF — возможность интеграции ИТ-инфраструктуры компании с облачными платформами Amazon EC2, OpenStack, RackSpace и Google Cloud. Если принять во внимание современные тенденции развития бизнеса, эта функция весьма актуальна.

С самого начала CHEF разрабатывался для управления Linux-компьютерами, но в настоящее время приложение можно использовать в сети с Windows-машинами. Таким образом, решение прекрасно подходит компаниям, только начинающим миграцию на открытые решения, позволяя сделать процесс постепенным.

Плюсы CHEF:

– Быстрота развертывания. Добавление в сеть новых устройств - простая и не требующая долгого времени операция.

– Гибкость настроек: Благодаря Bookshelfам, Chef позволяет создать несколько сценариев поведения сети, которые позволяют за короткое время переконфигурировать сеть оптимальным образом для выполнения определенного рода задач. Такая возможность актуальна для тех сетей, которые требуют быстрой адаптации под нужды компании. Оперативное перераспределение ресурсной мощности сети один из главных козырей данного решения

– Доступность: Любой интересующийся человек может скачать ознакомительную версию и попробовать писать свои рецепты, а потом уже приобрести лицензию и внедрить решения Chef непосредственно в рабочий процесс.

– Мультиплатформенность: Рецепты Chef можно адаптировать под любую операционную систему, и менять конфигурации ОС клиентских машин.

Минусы CHEF:

– Человеческий фактор: Если ошибиться в коде и применить некорректные настройки можно столкнуться с рядом проблем, от потери соединения до полной потери данных с выходом удаленного оборудования из строя.

– Безопасность: Если не обеспечить защиту сети должным образом, то проникновение в систему злоумышленника и перехват управления системой может привести к серьезному ущербу, особенно в сетях крупных корпораций.

– Громоздкость: Каждая строка настроек конфигурации должна быть выверена, и это требует от оператора особого внимания при создании и при проверке рецептов и кукбуков.

– Производительность: Данное решение на текущий момент несколько уступает конкурентам в производительности и потреблении ресурсов рабочей станции.

Ansible - новые ребята в DevOps направлении, по сравнению, например, с Chef или Puppet, но успели круто зарекомендовать себя в сообществе профессионалов за простоту и скоуп возможностей. Его playbooks понятны и легко читаемы, даже без особых знаний. Playbooks в Ansible это файл в формате YAML, который содержит последовательность состояния ресурсов системы, задач, которые позволяет запустить то или состояние сервера.

Ansible — система централизованного удалённого управления конфигурациями. Обычно применяется для обеспечения работы компьютеров, работающих на Linux, хотя Windows тоже поддерживается. Ansible был изначально проектом с открытым исходным кодом, который предназначался для автоматизации настройки и развертывания ПО в сетевых контурах компаний.

С 2015 г. разрабатывающая это решение компания принадлежит Red Hat. Таким образом, инструмент прекрасно подойдёт корпоративным пользователям, в основе ИТ-инфраструктуры которых лежит RHEL или его общедоступный аналог CentOS.

Ansible использует собственный язык программирования, специально адаптированный для решения задач управления процессом автоматизации и настройки ИТ-инфраструктуры компании. Для доступа к узлам из управляющей машины применяется SSH, что позволяет не устанавливать на клиентские машины никакого дополнительного ПО.

Привлекательные свойства Ansible: низкий порог вхождения, высокая надёжность и относительная простота. И, конечно же, в пользу этого инструмента говорит поддержка одной из ведущих Linux-компаний мира.

Плюсы Ansible:

– Легкость в изучении.

– Написан на Python. Количество людей, знающих Python гораздо больше, чем тех, кто

знает Ruby. Python библиотека, обычно, по умолчанию присутствует в любом Linux дистрибутиве, чего не сказать о Ruby. Ansible поддерживает написание модулей на любом языке программирования. Единственное требование – формат ответа должен быть JSON.

– Высокая надежность. Используя стандартный синтаксис Ansible Playbook, вы можете устанавливать правила брандмауэра, блокировать пользователей и группы, применять настраиваемые политики безопасности и автоматизировать другие функции безопасности.

– Не нужно ставить клиента (агента) на машину. Для управления узлами, Ansible обрабатывает все коммуникации между мастер – узлами и узлами – агентами по стандартному SSH, или через модуль Paramiko, который является частью Python SSH второй версии. Не нужно ставить агентское ПО на удаленные машины – только SSH подключение. Отсюда упрощение обслуживания и устранения неполадок.

– YAML плейбуки. Плейбуки в Ансибл невероятно просты и читаемы.

– Портал Ansible Galaxy. Это объединение Ansible сообщества, где люди делятся наработками и решениями той или иной задачи.

Минусы Ansible:

– Проблемы с интерфейсом (UI). Изначально Ansible разработан для работы с командной строкой. Ansible Tower дает возможность через GUI управлять Ansible, рисовать workflow и т.д. Но Tower позволяет делать только 85% рабочего функционала Ansible, который можно делать через командную строку. В добавок, конфигурации, внесенные через интерфейс, зачастую не синхронизируются с CLI – конфигами. Ansible Tower находится на стадии разработки.

– Нет работы с состоянием машин/процессов. Ansible не имеет понятия «состояние» и, соответственно, не отслеживает его. Он не смотрит на зависимости, а просто выполняет последовательный ряд задач/процессов.

– Слабая поддержка совместимости с Windows. Взаимодействие с Windows машинами происходит через PowerShell, и все равно требуется Linux хост для такой коммуникации.

– Новизна продукта. Меньше опыта, вследствие чего до сих пор выявляются баги.

Salt-Программное обеспечение имеет открытый исходный код и написано на языке Python, в чем имеет сходство с Ansible. Предусматривает механизм удаленной работы. Для выполнения команд используется push-модель и протокол SSH.

Отличается выполнением поставленных задач в высокой скорости, чем затмевает даже вышерассмотренные системы. Можно управлять конфигурацией, инфраструктурой и осуществлять оркестровку. Известно приложение с 2011 года.

Работать с приложением просто, как и настраивать его. Архитектура удаленного выполнения – единая. Есть поддержка всех видов языков файлами конфигурации системы.

Еще одна особенность SaltStack заключается в возможности параллельного выполнения команд на удаленных системах, что способствует ускорению процесса автоматизации. Приложение отличается простым интерфейсом, отлично сотрудничает с ОС BSD Unix, Windows, VMware vSphere.

Мощная структура управления SaltStack способна справиться с 10 000 машин. Это безопасный продукт, который использует зашифрованный протокол, имеет быструю и легкую коммуникационную шину Fast, что гарантирует основу для удаленного механизма выполнения задач. Процесс осуществляется автоматизировано, в этих целях используется облачный контроллер Salt Virt.

Возможности SaltStack Enterprise:

– оркестровка и автоматизация для CloudOps;

– автоматизация для ITOps;

– непрерывная интеграция и развертывание кода;

– мониторинг приложений и автоматическое восстановление;

– автоматизация рабочего процесса инструментария DevOps с поддержкой Puppet, Chef, Docker, Jenkins, Git и т. д.

Плюсы:

– Эффективен для высокой масштабируемости и отказоустойчивых сред.

– Простое и понятное использование после первоначальной установки и настройки.

– Сильный самоанализ.

– Активное сообщество и поддержка.

– Многофункциональный и согласованный синтаксис YAML для всех задач написания сценариев, при этом Python обеспечивает низкую кривую обучения для разработчиков.

Минусы:

– Процесс установки может быть нелегким для новых пользователей.

– Документация плохо управляется, и ее сложно проверять.

– Веб-интерфейс предлагает ограниченные возможности и функции.

– Это не лучший вариант для ОС, отличных от Linux.

– Платформа новая и не совсем зрелая по сравнению с Puppet и Chef.

Решения SaltStack SecOps обещают обеспечить «полную автоматизацию с обратной связью для соответствия ИТ операционным системам и инфраструктуре, а также устранение уязвимостей в своих локальных, облачных или гибридных средах». Этот инструмент обещает позволить вам определять настраиваемые политики на основе профилей соответствия отраслевым стандартам, таким как CIS и DISA STIGS, а затем запускать

непрерывное обнаружение и автономное применение политик на основе событий для выявления проблем, которые могут возникнуть.

Событийный характер SaltStack отличает его от других решений, описанных здесь. Однако это тоже потенциальная уязвимость. В любой архитектуре, управляемой событиями, невозможно предвидеть все возможные события, а когда дело касается безопасности, это означает, что в системе неизбежно присутствуют слабые места, которые хакеры могут обнаружить, а ваша группа безопасности – нет.

Заключение

В результате анализа сформирована таблица с кратким описанием положительных и отрицательных сторон каждого из инструментов.

Таблица 1 - Положительные и отрицательные стороны инструментов администрирования

	Puppet	Chef	Anisible	Salt
За	<ul style="list-style-type: none"> - Модули могут быть написаны на Ruby, или на более простом, производном от Ruby языке - Команды Push позволяют применять изменения немедленно - Веб-интерфейс поддерживает отчеты, инвентаризацию и управление узлами в реальном времени - Детализированные отчеты в работе агентов и конфигурация узлов 	<ul style="list-style-type: none"> - Использование всех возможностей Ruby - Централизованные, основанные на JSON массивы данных позволяют скриптам выполнять переносные во время работы - Веб-интерфейс позволяет вести поиск и учет узлов, просматривать их активность, применять хранилища данных и роли 	<ul style="list-style-type: none"> - Модули могут быть написаны почти на любом языке - Не требуются агенты на управляемых узлах - Веб-интерфейс позволяет настраивать пользователей, команды и оборудование, применять сценарии - Очень просто настраивается и запускается 	<ul style="list-style-type: none"> - Конфигурационные файлы могут быть простыми YAML-шаблонами или скриптами на Python и PyDSL - Может связываться с клиентами через SSH или с помощью локально установленных агентов - Веб-интерфейс позволяет просматривать запущенные задачи, статус подчиненных узлов и позволяет выполнить команды на клиентах - Крайне хорошо масштабируется
Против	<ul style="list-style-type: none"> - Требуется изучение встроенного языка или Ruby - Процессу установки не хватает отчетов об ошибках 	<ul style="list-style-type: none"> - Требуется знание Ruby - В данный момент не хватает функциональных команд Push - Документация местами неясная 	<ul style="list-style-type: none"> - Недостает поддержки клиентов для Windows - Веб-интерфейс автоматически не связывается с существующей установкой Ansible; данные должны быть импортированы 	<ul style="list-style-type: none"> - Веб-интерфейс не такой зрелый и полный как у конкурентов - Не хватает инструментов для детальных отчетов

Анализируя таблицу, можем прийти к выводу, что Salt – оптимальный вариант для админов, которые предпочитают гибкие системы, позволяющие управлять облачными инфраструктурами и обширной документацией. К тому же, благодаря реализации на Python он будет понятен для начинающий пользователей. Также приложение устроит тех, кто запланировал построить сеть, в которой легко будут существовать без сбоев самые разные окружения.

Литература

1. Лихачев С.В. Удаленное администрирование компьютеров на платформах Linux и Windows [Электронный ресурс] — URL: [https://cyberleninka.ru/article/n/udalennoe-](https://cyberleninka.ru/article/n/udalennoe-administrirovaniye-kompyuterov-na-platformah-linux-i-windows)

- administrirovaniye-kompyuterov-na-platformah-linux-i-windows (дата обращения: 10.11.2022)
2. Теоретические основы проектирования компьютерных сетей / Под ред. В.М. Вишневого. – М.: Техносфера, 2003 г. – 506 с. (дата обращения: 10.11.2022)
3. Документация Puppet [Электронный ресурс] — URL: <https://docs.puppet.com/> (дата обращения: 10.11.2022)
4. Библиотека решений Puppet [Электронный ресурс] — URL: <https://forge.puppet.com/> (дата обращения: 10.11.2022)
5. Документация Chef [Электронный ресурс] — URL: <https://docs.chef.io/> (дата обращения: 10.11.2022)

6. Библиотека пользовательских решений Chef [Электронный ресурс] — URL: <https://supermarket.chef.io/> (дата обращения: 10.11.2022)
7. Документация Ansible [Электронный ресурс] — URL: <http://docs.ansible.com/> (дата обращения: 10.11.2022)
8. Модули Ansible [Электронный ресурс] — URL: http://docs.ansible.com/ansible/list_of_all_modules.html (дата обращения: 10.11.2022)
9. Ansible tower [Электронный ресурс] — URL: <https://www.ansible.com/tower> (дата обращения: 10.11.2022)
10. Документация SaltStack [Электронный ресурс] — URL: <https://docs.saltstack.com/> (дата обращения: 10.11.2022)
11. Библиотека решений SaltStack для Windows [Электронный ресурс] — URL: <https://docs.saltstack.com/en/latest/topics/windows/windows-package-manager.html> (дата обращения: 10.11.2022)

References

1. Lihachev S.V. Udalennoe administrirovanie kompyuterov na platformah Linux i Windows [Elektronnyj resurs] — URL: <https://cyberleninka.ru/article/n/udalennoe-administrirovanie-kompyuterov-na-platformah-linux-i-windows> (data obrashcheniya: 10.11.2022)
2. Teoreticheskie osnovy proektirovaniya kompyuternyh setej / Pod red. V.M. Vishnevskogo. — M.: Tekhnosfera, 2003 g. — 506 s. (data obrashcheniya: 10.11.2022)
3. Dokumentaciya Puppet [Elektronnyj resurs] — URL: <https://docs.puppet.com/> (data obrashcheniya: 10.11.2022)
4. Biblioteka reshenij Puppet [Elektronnyj resurs] — URL: <https://forge.puppet.com/> (data obrashcheniya: 10.11.2022)
5. Dokumentaciya Chef [Elektronnyj resurs] — URL: <https://docs.chef.io/> (data obrashcheniya: 10.11.2022)
6. Biblioteka pol'zovatel'skih reshenij Chef [Elektronnyj resurs] — URL: <https://supermarket.chef.io/> (data obrashcheniya: 10.11.2022)
7. Dokumentaciya Ansible [Elektronnyj resurs] — URL: <http://docs.ansible.com/> (data obrashcheniya: 10.11.2022)
8. Moduli Ansible [Elektronnyj resurs] — URL: http://docs.ansible.com/ansible/list_of_all_modules.html (data obrashcheniya: 10.11.2022)
9. Ansible tower [Elektronnyj resurs] — URL: <https://www.ansible.com/tower> (data obrashcheniya: 10.11.2022)
10. Dokumentaciya SaltStack [Elektronnyj resurs] — URL: <https://docs.saltstack.com/> (data obrashcheniya: 10.11.2022)
11. Biblioteka reshenij SaltStack dlya Windows [Elektronnyj resurs] — URL: <https://docs.saltstack.com/en/latest/topics/windows/windows-package-manager.html> (data obrashcheniya: 10.11.2022)

УДК 004.621.396.932

DOI: 10.34046/aumsuomt105/42

ОБЕСПЕЧЕНИЕ ПОМЕХОЗАЩИЩЕННОСТИ ПЕРЕДАЧИ ДАННЫХ ПО РАДИОЛОКАЦИОННЫМ КАНАЛАМ СВЯЗИ

И.М. Данцевич, кандидат технических наук, доцент

М.Н. Лютикова, кандидат технических наук, доцент

Ж.Ж. Жумаев, доктор технических наук, профессор

В статье исследованы способы формирования адаптивного алгоритма кодирования 8/16 бит на бод сигналов управления или телемеханики с заданными критериями качеств в условиях физически нерасширяемого канала передачи данных. Совмещение систем пеленга и информационного обмена в одной радиолокационной станции, сталкивается с трудностями формирования передаваемого сообщения. Необходимо использовать помехозащищенные методы сверточного кодирования, чтобы в полосе ЛЧМ зондирующего и отраженного сигнала детектировать передаваемые биты сообщений. Формирование широкополосного сигнала с использованием дробного преобразования Фурье позволяет формировать протоколы сообщений мультиплексированием с ортогональным частотным разделением. Предлагаемый способ кодирования позволяет реализовать помехоустойчивое кодирование в соотношении 8 или 16 бит на бод. Побережье большинства стран обустроено по требованиям безопасности мореплавания и имеют достаточное количество совмещенных береговых радиолокационных постов управления, что позволяет реализовать каботажное беспилотное мореплавание. Применение радиосвязи в полосе навигационного радара позволяет реализовать концепцию совмещенной интеллектуальной транспортной системы. Задачами такой системы являются определение позиции (пеленг) наблюдаемого морского подвижного объекта (МПО), а также запрос и ответ телеметрии параметров движения МПО, передачи с борта МПО навигационных данных с радара беспилотного объекта, а также передача криптозащищенных команд с берегового пункта управления.

Ключевые слова: телеметрия, кодирование, широкополосный сигнал, амплитудно-фазовая модуляция, квадратурная фазовая модуляция, функция Бесселя первого рода, линейно-частотное модулирование, ортогонально частотное разделение, динамические протоколы.