



РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ OPEN-SOURCE ГИПЕРВИЗОРА KVM ПРИ СОЗДАНИИ СОВРЕМЕННОЙ ПЛАТФОРМЫ ВИРТУАЛИЗАЦИИ

*Степанов Николай Викторович,
Sitronics Group, г. Москва*

E-mail: nklstpnv@outlook.com

Аннотация. В статье рассматривается расширение возможностей open-source гипервизора KVM при создании современных платформ виртуализации. Осуществлен анализ архитектуры и ключевых характеристик KVM, а также предложены методы улучшения производительности, безопасности и масштабируемости. Рассмотрены актуальные проблемы использования KVM. В статье также предложены решения для расширения функционала KVM. Представлены методы тестирования и оценки эффективности внедренных решений в реальных условиях. Работа направлена на повышение эффективности и гибкости использования KVM в качестве гипервизора для облачных и высокопроизводительных вычислительных систем.

Ключевые слова: KVM, виртуализация, гипервизор, масштабируемость, производительность, безопасность, NUMA, GPU-пассинг, контейнеризация, OpenStack, Ceph, виртуализированные устройства, сетевая виртуализация.

Актуальность исследования

С развитием технологий информационных систем и требованием к обеспечению высокой гибкости и масштабируемости в сфере облачных вычислений, виртуализация становится одной из основополагающих технологий. Виртуализация позволяет эффективно использовать ресурсы физических машин, а также предоставляет гибкость в управлении вычислительными мощностями. В этом контексте гипервизор KVM (Kernel-based Virtual Machine), являющийся open-source решением, представляет собой одну из наиболее популярных платформ для виртуализации в операционных системах на базе Linux.

Однако несмотря на свою высокую производительность и широкое распространение, KVM сталкивается с рядом ограничений, связанных с масштабируемостью, производительностью, а также с недостаточной поддержкой новейших технологий, таких как виртуализация с использованием

графических процессоров (GPU) и интеграция с облачными платформами. В условиях стремительного роста требований к вычислительным мощностям и обеспечению высокой степени изоляции виртуальных машин, стоит задача расширения функциональности KVM для эффективной поддержки современных инфраструктур виртуализации и облачных решений.

Актуальность исследования обусловлена необходимостью развития и оптимизации гипервизора KVM для соответствия новым требованиям виртуализационных платформ и расширения его функциональных возможностей для решения комплексных задач в области высокопроизводительных вычислений, облачных сервисов и безопасности.

Цель исследования

Целью данного исследования является разработка методов расширения возможностей гипервизора KVM с целью улучшения его производительности, безопасности и масштабируемости при создании современных платформ виртуализации.

Материалы и методы исследования

Материалы исследования: научные статьи, книги и публикации в области компьютерных технологий, а также документация по KVM и связанным с ним технологиям (virtio, NUMA, OpenStack).

Методы исследования: теоретический анализ архитектуры и характеристик гипервизора KVM, сравнительный анализ существующих решений в области виртуализации.

Результаты исследования

Виртуализация – это технология, позволяющая создать несколько логических (виртуальных) систем на основе одной физической машины, каждый из которых функционирует как отдельная вычислительная система, с доступом к своим собственным ресурсам. Принцип виртуализации состоит в абстрагировании аппаратного обеспечения, что позволяет эффективно управлять ресурсами системы и предоставлять их виртуальным машинам (ВМ), с их независимой работой, защитой и изоляцией.

Основными принципами виртуализации являются:

Изоляция: каждая виртуальная машина работает в своей собственной среде, что предотвращает воздействие одной ВМ на другую, даже в случае ошибок или сбоев.

Абстракция: виртуальная машина абстрагирует и «маскирует» ресурсы, доступные на физическом уровне (ЦП, память, диск и т.д.), предоставляя виртуализированный интерфейс.

Консолидирование: на одной физической машине может быть размещено несколько виртуальных машин, что способствует более эффективному использованию вычислительных ресурсов.

Мобильность и управление: виртуальные машины могут быть легко перенесены на другую физическую машину, что дает гибкость в управлении инфраструктурой.

Технологии виртуализации включают не только серверную виртуализацию, но и виртуализацию хранения данных, сетевых ресурсов, а также создание виртуальных рабочих станций и инфраструктур виртуальных приложений [1, с. 107]. Эти технологии используют гипервизоры, которые являются важным звеном между физическим оборудованием и виртуальными машинами.

Гипервизор (или виртуализатор) – это программное обеспечение, которое управляет виртуальными машинами, предоставляя им доступ к аппаратным ресурсам. В зависимости от того, как гипервизор взаимодействует с физическим оборудованием и операционной системой, гипервизоры делятся на два типа: тип 1 и тип 2.

Гипервизор типа 1 (bare-metal) – это гипервизор, который работает непосредственно на физическом оборудовании, без необходимости использования хостовой операционной системы. Он управляет всеми ресурсами компьютера и предоставляет виртуальные машины для работы на этом оборудовании.

Пример гипервизора типа 1 – это KVM, VMware ESXi, Microsoft Hyper-V. Преимуществом гипервизора типа 1 является высокая производительность и безопасность, так как его работа не зависит от хостовой операционной системы, и он имеет прямой доступ к аппаратным ресурсам.

Гипервизор типа 2 (hosted) – это гипервизор, который работает как обычное приложение в хостовой операционной системе. Он использует ресурсы хостовой ОС для управления виртуальными машинами.

Примеры гипервизоров типа 2 – это VMware Workstation, Oracle VirtualBox, Parallels Desktop. Гипервизоры типа 2 часто применяются в условиях рабочего стола для виртуализации отдельных приложений или операционных систем, но их производительность обычно ниже по сравнению с гипервизорами типа 1, так как они зависят от работы хостовой операционной системы.

Сравнение гипервизоров типов 1 и 2 представлено в таблице 1.

Таблица 1

Сравнение гипервизоров типов 1 и 2

Характеристика	Гипервизор типа 1	Гипервизор типа 2
Местоположение	Работает непосредственно на железе	Работает в рамках хостовой ОС
Производительность	Высокая, минимум накладных расходов	Ниже из-за зависимостей от ОС
Безопасность	Высокая, изолирован от хостовой ОС	Зависит от хостовой ОС
Применение	Серверы, облачные платформы	Личные компьютеры, тестирование

Open-source решения для виртуализации, такие как KVM, Xen, и другие, привлекают внимание благодаря своей гибкости, возможности модификации и низкой стоимости. В таблице 2 приведены основные преимущества и недостатки использования open-source гипервизоров для создания виртуализированных платформ.

Таблица 2

Основные преимущества и недостатки
использования open-source решений для виртуализации

Характеристика	Преимущества	Недостатки
Стоимость	Нет лицензионных сборов	Зависимость от собственного опыта и знаний
Гибкость	Высокая кастомизация и интеграция	Сложность настройки и администрирования
Сообщество и поддержка	Активные сообщества, быстрые обновления	Отсутствие официальной коммерческой поддержки
Безопасность	Открытый код позволяет проводить аудит безопасности	Требуется постоянное внимание к уязвимостям
Функциональность	Поддержка множества технологий и платформ	Ограниченная поддержка некоторых новых технологий
Совместимость	Поддержка различных аппаратных платформ	Интеграция с облачными и гибридными решениями может быть сложной

KVM – это open-source гипервизор, который использует возможности ядра операционной системы Linux для виртуализации. Он был разработан как часть ядра Linux и является одним из наиболее популярных решений для виртуализации в Linux-средах [3, с. 119].

Идея создания KVM возникла в середине 2000-х годов, когда потребность в эффективной виртуализации для Linux-систем стала все более очевидной. В 2007 году Кристиан Кох и других разработчиков из компании Qumranet, которая в дальнейшем была приобретена компанией Red Hat, предложили использовать возможности процессоров с поддержкой аппаратной виртуализации, таких как Intel VT-x и AMD-V. KVM был интегрирован в ядро Linux в 2007 году, начиная с версии 2.6.20, что значительно улучшило производительность виртуализации за счет использования этих технологий.

Первые версии KVM использовали базовые возможности виртуализации, а в последующие годы система претерпела значительные улучшения, включая расширение поддержки различных архитектур, улучшение производительности, повышение совместимости с различными инструментами управления виртуальными машинами, такими как libvirt, и добавление новых функций, например, поддержка NUMA для работы с большим количеством виртуальных машин.

К 2010 году KVM стал одним из ключевых гипервизоров для Linux и получил широкое распространение в коммерческих и облачных решениях. С развитием таких проектов, как OpenStack и oVirt, KVM занял важную нишу в инфраструктуре виртуализации с открытым исходным кодом.

Архитектура гипервизора KVM отличается своей тесной интеграцией с ядром Linux, что позволяет ему эффективно использовать функции операционной системы, такие как управление памятью, планирование задач и обработку прерываний. В отличие от гипервизоров типа 1, которые работают непосредственно на аппаратном уровне, KVM функционирует как часть операционной системы, что позволяет значительно снизить накладные расходы на управление ресурсами.

Основные компоненты KVM:

Модуль ядра KVM – это основной компонент гипервизора. Он предоставляет интерфейсы для использования функций аппаратной виртуализации, таких как виртуализация процессоров и управления памятью. Модуль KVM использует возможности процессоров, поддерживающих технологии Intel VT-x и AMD-V, для создания изолированных виртуальных машин.

Qemu (Quick Emulator) – это пользовательский инструмент, который работает в тесной связке с KVM для управления виртуальными машинами. QEMU эмулирует аппаратные ресурсы, предоставляя виртуальным машинам доступ к необходимым устройствам, таким как сетевые карты, диски и другие ресурсы. В сочетании с KVM, QEMU предоставляет высокую производительность за счет использования аппаратной виртуализации.

Libvirt – это библиотека и набор инструментов, предоставляющих абстракцию для управления виртуальными машинами. Libvirt служит интерфейсом для взаимодействия с KVM и предоставляет удобные инструменты для создания, удаления, управления и мониторинга виртуальных машин. Libvirt также поддерживает взаимодействие с другими гипервизорами, что позволяет интегрировать KVM в более крупные системы управления виртуализацией, такие как OpenStack.

Архитектура KVM требует, чтобы ядро Linux поддерживало функции виртуализации, и использует стандартные механизмы управления процессами и памятью, что дает большую гибкость и простоту в управлении виртуализированными средами. Взаимодействие между гипервизором и операционной системой осуществляется через интерфейсы, такие как KVM API, которые позволяют организовывать работу виртуальных машин.

KVM использует гибридный подход к виртуализации, сочетая возможности аппаратной виртуализации с традиционными методами программной виртуализации. Важным аспектом работы KVM является использование аппаратных средств, таких как технологии Intel VT-x и AMD-V, которые предоставляют виртуализатору прямой доступ к процессорным инструкциям, что значительно повышает производительность и эффективность.

Ключевые особенности работы KVM:

Аппаратная виртуализация: KVM использует аппаратные расширения процессора для создания виртуальных машин с минимальными накладными расходами. Эти технологии позволяют виртуальным машинам работать практически без потери производительности, поскольку они могут использовать функции виртуализации процессора, такие как изоляция памяти и виртуализация контекстов.

Поддержка многозадачности: виртуальные машины в KVM управляются ядром Linux, которое может использовать возможности планировщика задач для эффективного распределения времени процессора между виртуальными машинами.

Поддержка различных архитектур: KVM поддерживает различные архитектуры, включая x86, ARM и PowerPC, что делает его универсальным решением для виртуализации как в облачных, так и в частных инфраструктурах.

Интеграция с другими компонентами виртуализации: KVM интегрируется с рядом инструментов и проектов для более удобного управления виртуализацией. Например, с OpenStack для создания частных облаков и с oVirt для управления виртуальными машинами в дата-центрах.

В последние годы KVM значительно улучшил свои возможности в области производительности, безопасности и масштабируемости, что позволило ему стать одним из самых эффективных решений для виртуализации в облачных и высокопроизводительных вычислительных средах [4, с. 90].

Несмотря на свою популярность и широкое распространение, гипервизор KVM имеет ряд проблем и ограничений, которые могут существенно повлиять на его эффективность и использование в различных виртуализированных платформах (таблица 3).

Для повышения производительности, безопасности и функциональности гипервизора KVM, разработаны различные методы расширения его возможностей. Эти методы направлены на улучшение функционала виртуализации, повышение совместимости с новыми аппаратными и программными технологиями, а также на упрощение управления виртуализированными средами. Рассмотрим ключевые направления расширения возможностей KVM:

1. Поддержка более эффективного ввода-вывода (I/O): Виртуальные машины, работающие на KVM, могут столкнуться с проблемами производительности при обработке ввода-вывода, особенно в случаях высоконагруженных приложений. Для решения этой проблемы используются

такие технологии, как virtio (виртуализированные устройства ввода-вывода), которые значительно сокращают накладные расходы, связанные с эмуляцией аппаратных устройств. Кроме того, для улучшения производительности на уровне хранения данных применяются решения, такие как virtio-blk для блочных устройств и virtio-net для сетевых устройств (рис. 1).

Таблица 3

Проблемы и ограничения KVM в контексте создания платформы виртуализации

Проблема / Ограничение	Описание	Возможные решения / Направления развития
Масштабируемость	KVM может столкнуться с проблемами при управлении очень большим количеством виртуальных машин, особенно в условиях ограниченных аппаратных ресурсов	Оптимизация планировщика задач, улучшение поддержки NUMA, использование технологий контейнеризации
Поддержка новых технологий	KVM может отставать от новых аппаратных технологий, таких как новые процессоры, устройства или алгоритмы виртуализации	Регулярные обновления ядра и драйверов, улучшенная интеграция с новыми процессорами и аппаратными средствами
Производительность при высоких нагрузках	При запуске большого числа виртуальных машин производительность может снижаться из-за накладных расходов на управление памятью и процессами	Использование технологий аппаратной виртуализации, улучшение I/O-подсистемы и поддержка GPU-пассинга
Управление хранилищем данных	KVM может испытывать трудности с эффективным управлением большими объемами данных, особенно в распределённых системах	Интеграция с современными системами хранения данных (например, Ceph), улучшение поддержки виртуальных дисков
Интеграция с облачными платформами	KVM требует специфической настройки для интеграции с облачными решениями, такими как OpenStack или Kubernetes	Развитие официальных инструментов для интеграции с облачными платформами, улучшение совместимости
Безопасность	Несмотря на наличие встроенных механизмов безопасности, KVM может быть уязвимым к атакам на уровне гипервизора или из-за недостатков в виртуализационных драйверах	Разработка новых методов защиты, использование аппаратных средств безопасности, таких как SEV и TXT
Сложность управления и настройки	Для большинства пользователей KVM требует сложной настройки, особенно в сочетании с другими компонентами виртуализации (например, libvirt или OpenStack)	Разработка инструментов для упрощения администрирования и мониторинга виртуализированных систем
Поддержка сетевых решений	Виртуализация сетевых ресурсов (например, поддержка SDN, виртуальных маршрутизаторов) в KVM иногда ограничена	Развитие решений для виртуализации сети, интеграция с более сложными сетевыми технологиями
Поддержка графических процессоров (GPU)	Ограниченная поддержка GPU-пассинга и виртуализации графических ресурсов, что ограничивает использование KVM в задачах, требующих интенсивных вычислений на GPU	Внедрение улучшенных решений для GPU-пассинга, поддержка технологий виртуализации GPU от производителей (например, NVIDIA vGPU)
Совместимость с существующими решениями	KVM может сталкиваться с проблемами совместимости при миграции виртуальных машин из других платформ виртуализации (например, VMware или Hyper-V)	Разработка инструментов для миграции между гипервизорами, улучшение совместимости с различными форматами виртуальных машин

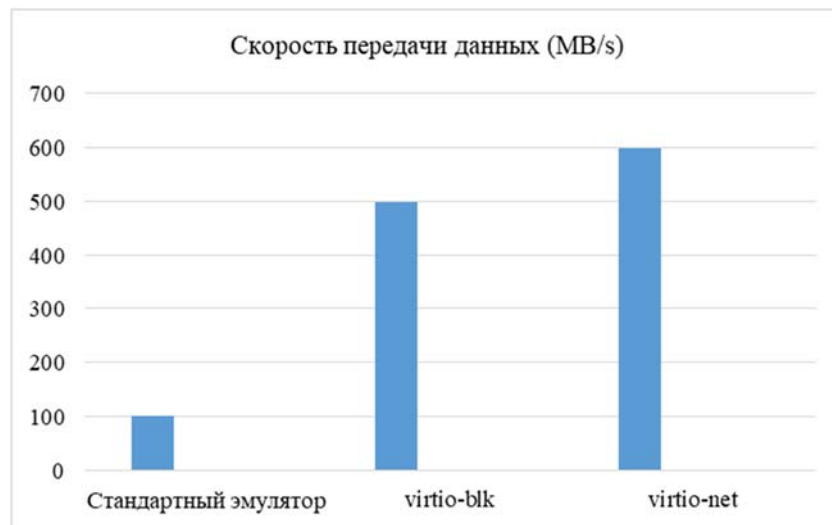


Рис. 1 Сравнение производительности ввода-вывода с использованием разных технологий виртуализации

Также важным направлением является интеграция с современными системами хранения данных, например, с Ceph и GlusterFS, которые позволяют оптимизировать работу с большими объемами данных в распределённых и облачных системах.

2. Поддержка NUMA: Технология NUMA используется для управления памятью в многопроцессорных системах, когда процессоры имеют доступ к памяти с разной задержкой в зависимости от их местоположения в системе. Для эффективного использования таких систем, KVM был дополнен поддержкой NUMA, что позволяет виртуальным машинам работать с распределённой памятью, улучшая производительность и масштабируемость.

Для визуализации эффективности использования NUMA, можно провести сравнение двух линий: одна для системы без поддержки NUMA, и другая для системы с поддержкой NUMA (рис. 2).

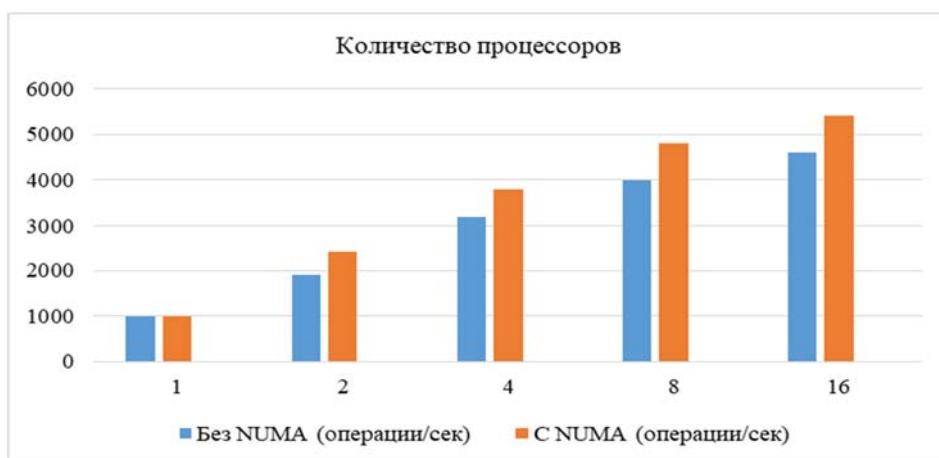


Рис. 2 Эффективность использования NUMA в многопроцессорных системах

3. Поддержка GPU-пассинга. Для задач, требующих интенсивных вычислений на графических процессорах (например, для глубокого обучения или высокопроизводительных вычислений), KVM предоставляет возможность использования GPU-пассинга (рис. 3). Это позволяет выделить один или несколько графических процессоров для использования конкретной виртуальной машиной, обеспечивая доступ к мощностям GPU для ускоренных вычислений [2, с. 435].

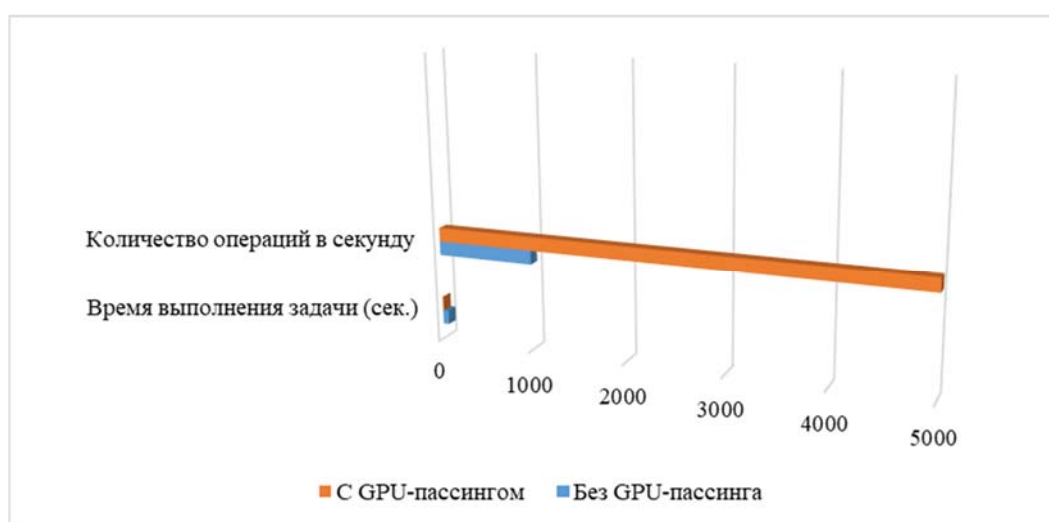


Рис. 3 Производительность при использовании GPU-пассинга

4. Поддержка контейнеризации (KVM и контейнеры): В последние годы наблюдается рост популярности контейнеризации, благодаря таким решениям как Docker и Kubernetes. Для объединения преимуществ виртуализации и контейнеризации был разработан гибридный подход, при котором KVM используется в сочетании с контейнерами. В этой архитектуре виртуальная машина может работать как изолированная среда с полноценным операционным окружением, в то время как контейнеры обеспечивают более легковесную изоляцию для микросервисных приложений.

5. Использование технологий безопасности: KVM также активно развивает функционал безопасности, предлагая ряд технологий для защиты виртуализированных систем. Среди них – SELinux и AppArmor, которые обеспечивают контроль доступа на уровне операционной системы. KVM также поддерживает sVirt, технологию, которая использует механизм безопасности SELinux для изоляции виртуальных машин, улучшая защиту данных и приложений от атак на гипервизор.

Для повышения безопасности KVM используется интеграция с технологиями Intel TXT (Trusted Execution Technology), AMD SEV (Secure Encrypted Virtualization) и т.д. (рис. 4).

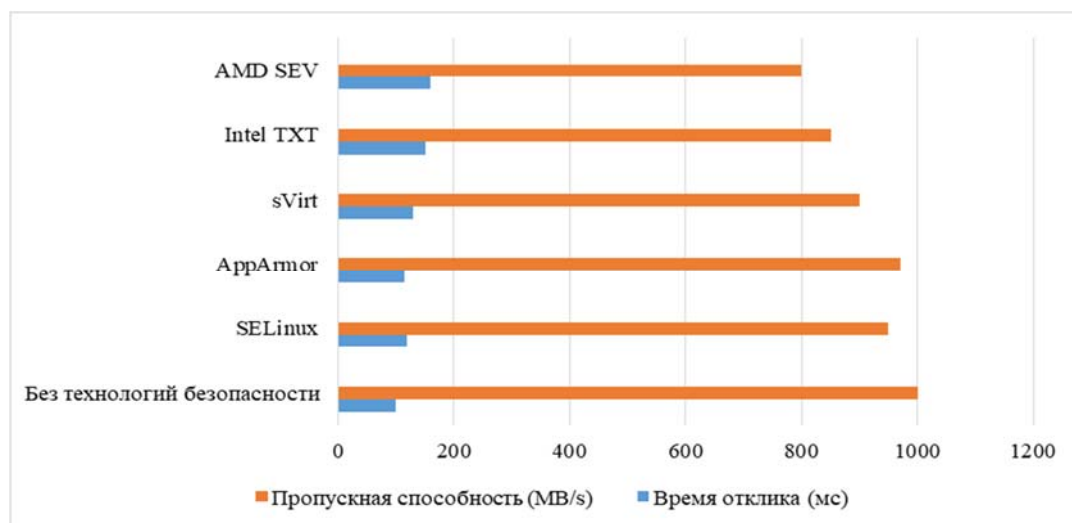


Рис. 4. Влияние технологий безопасности на производительность

Эти технологии помогают защищать данные и приложения в виртуализированных средах, обеспечивая шифрование и изоляцию данных на аппаратном уровне.

Выводы

В ходе проведенного исследования были разработаны методы расширения возможностей гипервизора KVM с целью улучшения его производительности, безопасности и масштабируемости при создании современных платформ виртуализации. На основе теоретического анализа архитектуры KVM и существующих решений в области виртуализации предложены конкретные улучшения, направленные на оптимизацию работы гипервизора в облачных и высокопроизводительных вычислительных системах. Таким образом, данная работа способствует дальнейшему развитию и улучшению open-source решений в области виртуализации, а также может послужить основой для дальнейших исследований и улучшений в сфере гипервизоров.

Литература:

1. Дорошенко В.С., Шадрин Д.Б. Использование виртуальных машин в обучении // Новые образовательные технологии в вузе. – 2015. – С. 106-108.
2. Кудрявцев А.О., Кошелев В.К., Избышев А.О., Аветисян А.И. Высокопроизводительные вычисления как облачный сервис: ключевые проблемы // Параллельные вычислительные технологии 2013 (ПаВТ'2013). – 2013. – С. 432-438.
3. Стасьев Д.О. Контроль целостности компонентов виртуальных машин, созданных на базе гипервизора KVM // Безопасность информационных технологий. – 2020. – Т. 27, № 2. – С. 118-131.
4. Шевчук М.В., Костякова В.Г. Облачные платформы и технологии виртуализации в образовании // Педагогическое образование и наука. – 2022. – № 6. – С. 88-92.