DOI: 10.58168/PPSPM2025_82-89
УДК 004.942

СРАВНИТЕЛЬНЫЙ АНАЛИЗ DOCKER И GVISOR ДЛЯ ИЗОЛЯЦИИ WORKLOADS. КОНТЕЙНЕРИЗАЦИЯ И ВИРТУАЛИЗАЦИЯ

А.С. Кравченко, А.М. Тюнина

Воронежский государственный лесотехнический университет
им. Г.Ф. Морозова

Аннотация: в статье проведен сравнительный анализ архитектурных подходов Docker и gVisor с оценкой компромиссов между безопасностью, производительностью и совместимостью. Методология включает тестирование производительности с использованием стандартных бенчмарков, анализ поверхности атаки и оценку совместимости приложений. Выявлена ограниченная совместимость gVisor с приложениями, интенсивно использующими специфические системные вызовы. Практическая значимость работы заключается в разработке критериев выбора технологии изоляции в зависимости от требований безопасности и производительности.

Ключевые слова: контейнеризация, изоляция рабочих нагрузок, Docker, gVisor, безопасность, производительность, поверхность атаки.

SRAVNITEL'NYY ANALIZ DOCKER I GVISOR DLYA IZOLYATSII WORKLOADS. KONTEYNERIZATSIYA I VIRTUALIZATSIYA

A.S. Kravchenko, A.M. Tyunina

Voronezh State University of Forestry and Technologies named after G.F. Morozov

Annotatsiya: v stat'ye proveden sravnitel'nyy analiz arkhitekturnykh podkhodov Docker i gVisor s otsenkoy kompromissov mezhdu bezopasnost'yu, proizvoditel'nost'yu i sovmesti-most'yu. Metodologiya vklyuchayet testirovaniye proizvoditel'nosti s ispol'zovaniyem stan-dartnykh benchmarkov, analiz poverkhnosti ataki i otsenku sovmestimosti prilozheniy. Vyyavlena ogranichennaya sovmestimost' gVisor s prilozheniyami, intensivno ispol'zuyushchi-mi

spetsificheskiye sistemnyye vyzovy. Prakticheskaya znachimost' raboty zaklyuchayetsya v raz-rabotke kriteriyev vybora tekhnologii izolyatsii v zavisimosti ot trebovaniy bezopasnosti i proizvoditel'nosti.

Keywords: containerization, workload isolation, Docker, gVisor, security, performance, attack surface.

Актуальность проблемы обеспечения безопасной изоляции рабочих нагрузок в облачных средах постоянно возрастает в связи с массовым переходом на контейнеризированную архитектуру приложений. Современные облачные платформы обрабатывают тысячи изолированных рабочих нагрузок на одном физическом хосте, где компрометация одного контейнера может привести к нарушению безопасности всей системы. Традиционные решения контейнеризации, такие как Docker, используют общее ядро операционной системы для всех контейнеров, что создает значительную поверхность атаки и потенциально позволяет эскалация привилегий в случае уязвимостей в ядре ОС. В качестве альтернативы разрабатываются решения с усиленной изоляцией, среди которых gVisor от Google представляет особый интерес благодаря архитектуре, исключающей прямой доступ к системным вызовам ядра.

Обзор существующих исследований показывает противоречивость оценок эффективности различных подходов к изоляции. Работы, посвященные анализу производительности контейнеров, в основном фокусируются на сравнении Docker с традиционной виртуализацией, в то время как исследования безопасности часто ограничиваются теоретическим анализом уязвимостей. Недостаточно изученными остаются практические аспекты компромиссов между безопасностью и производительностью при использовании архитектур типа gVisor в реальных сценариях. Особую сложность представляет количественная оценка безопасности различных решений и ее корреляция с производительностью.

Целью данной работы является комплексный сравнительный анализ технологий Docker и gVisor для изоляции рабочих нагрузок с фокусом на оценке компромиссов между тремя ключевыми характеристиками: безопасностью, производительностью и совместимостью. Для достижения поставленной цели определены следующие задачи: формализация архитектурных различий между решениями, разработка методологии сравнительной оценки, экспериментальное измерение показателей производительности и безопасности, анализ совместимости с типовыми

рабочими нагрузками, выработка практических рекомендаций по выбору технологии изоляции.

Объектом исследования в данной работе являются технологии изоляции рабочих нагрузок Docker и gVisor, рассматриваемые как системы обеспечения безопасного выполнения приложений в разделяемой облачной среде. Docker представляет традиционный подход контейнеризации с использованием пространств имен ядра Linux и групп управления, обеспечивающий изоляцию на уровне операционной системы. gVisor реализует альтернативную архитектуру с изоляцией на уровне пользовательского пространства ядра, где системные вызовы приложений перехватываются и обрабатываются через прослойку - так называемое "гостевое ядро", написанное на Go.

Входными параметрами анализа являются характеристики рабочих нагрузок, включая интенсивность системных вызовов, профиль использования ресурсов, требования к сетевому взаимодействию и специфические зависимости от возможностей ядра. Дополнительными входными параметрами выступают конфигурации безопасности, такие как используемые механизмы изоляции, настройки политик доступа и параметры ограничения ресурсов. Для экспериментальной части критически важны метрики производительности, включая время выполнения операций, потребление ресурсов и латентность.

Выходными параметрами исследования являются количественные оценки безопасности, измеряемые через размер поверхности атаки и устойчивость к эскалации привилегий; показатели производительности, включая накладные расходы на изоляцию и масштабируемость; а также оценка совместимости, определяемая как способность выполнять различные типы приложений без модификации. Ключевой выходной метрикой является интегральный показатель эффективности, отражающий баланс между безопасностью и производительностью.

Ограничения исследования включают специфику используемого аппаратного обеспечения, версий программного обеспечения и выбранных рабочих нагрузок. Моделирование проводится для типовых сценариев использования в облачных средах, что может не полностью отражать специализированные случаи эксплуатации. Кроме того, оценка безопасности основывается на анализе поверхности атаки и исторических данных об уязвимостях, что не учитывает потенциальные неизвестные уязвимости. Важным ограничением является также фокус на Linux-based решениях, что исключает из рассмотрения другие операционные системы.

Для сравнительного анализа технологий изоляции Docker и gVisor разработана комплексная методология, сочетающая экспериментальные измерения и аналитическое моделирование. Основу методологии составляет концепция поверхности атаки, формализуемая как множество потенциальных векторов компрометации системы. Для Docker поверхность атаки определяется как объединение системных вызовов ядра Linux, доступных из пространства имен контейнера, и компонентов демона Docker. В случае gVisor поверхность атаки включает интерфейс системных вызовов гостевого ядра и механизмы взаимодействия с хост-системой. Количественная оценка безопасности вычисляется через метрику $S = 1 - \left(\frac{A}{R}\right)$, где A — количество доступных векторов атаки, R — общее количество векторов в эталонной системе.

Производительность оценивается через систему уравнений, описывающих накладные расходы на изоляцию. Для Docker временные затраты определяются как $T_d = T_b + T_n + T_c$, где T_b — базовое время выполнения, T_n — накладные расходы пространств имен, T_c — затраты на контрольные группы. Для gVisor модель усложняется: $T_g = T_b + T_s + T_v + T_m$, где T_s — затраты на перехват системных вызовов, T_v — верификация параметров, T_m — маршрутизация через гостевое ядро. Отношение производительности выражается как $R_{ratio} = \frac{T_d}{T_g}$.

Экспериментальная проверка включает три категории тестов: микротесты системных вызовов с использованием специально разработанных бенчмарков, макротесты производительности с использованием стандартных инструментов и тесты совместимости на наборе типовых приложений. Для системных вызовов измеряется время выполнения 1000 итераций операций файловой системы, сетевого взаимодействия и управления процессами. Производительность оценивается через серию тестов включающих вычисления, операции ввода-вывода и сетеую пропускную способность. Совместимость проверяется через способность выполнения репрезентативных приложений без модификации их кода.

Экспериментальная инфраструктура развернута на идентичных виртуальных машинах с 8 vCPU, 32 ГБ ОЗУ и SSD-дисками. Используется ОС Ubuntu 22.04 LTS с ядром 5.15, Docker 23.0 и gVisor 20230612. Нагрузка генерируется с помощью Apache Bench, Siege и собственных бенчмарков.

Измерения проводятся при различной интенсивности нагрузки от 100 до 10000 concurrent connections.

Результаты измерения поверхности атаки показывают значительное преимущество gVisor. Если Docker предоставляет доступ к 68% системных вызовов ядра Linux, то gVisor ограничивается 23%. Количество потенциальных векторов атаки через компоненты демона составляет 12 для Docker против 3 для gVisor. Интегральная оценка безопасности составляет 0.31 для Docker и 0.74 для gVisor по шкале от 0 до 1.

Производительность демонстрирует обратную картину. В тестах вычислений Docker показывает отставание в 3-5% относительно нативного выполнения, в то время как gVisor добавляет 15-20% накладных расходов. Операции файловой системы выявляют более существенный разрыв: при последовательном чтении Docker отстает на 8%, gVisor — на 42%. Сетевые тесты показывают максимальную разницу: пропускная способность TCP-соединений в Docker составляет 94% от нативной, в gVisor — лишь 56%. Задержки распределяются аналогично — 1.2 мс для Docker против 2.8 мс для gVisor при размере пакета 1 КБ.

Тесты совместимости выявляют 97% успешных запусков для Docker против 83% для gVisor. Наиболее проблемными для gVisor оказываются приложения, использующие расширенные системные вызовы ptrace, ioperm и специальные функции сетевых стеков. Приложения с стандартным POSIX-поведением демонстрируют полную совместимость с обеими системами. Для наглядности данные представлены в таблице 1.

Полученные результаты демонстрируют фундаментальный компромисс между безопасностью и производительностью при выборе технологии изоляции рабочих нагрузок. Преимущество gVisor в безопасности объясняется архитектурными особенностями — использование прослойки пользовательского пространства ядра эффективно ограничивает поверхность атаки. Однако эта архитектура становится источником существенных накладных расходов, особенно заметных в операциях ввода-вывода и сетевого взаимодействия. Анализ показывает, что дополнительные проверки системных вызовов и маршрутизация через гостевое ядро увеличивают латентность операций в среднем на 65-130% по сравнению с Docker.

Важным аспектом является зависимость производительности от типа рабочей нагрузки. Вычислительно интенсивные задачи демонстрируют относительно умеренное падение производительности в gVisor (15-20%), в то

время как приложения с активным сетевым взаимодействием и частыми операциями с файловой системой теряют до 45% производительности. Это объясняется архитектурными ограничениями гостевого ядра gVisor, которое не может полностью оптимизировать работу с системными ресурсами.

Таблица 1 – Сравнительный анализ Docker и gVisor

Критерий оценки	**Показатель**	**Docker**	**gVisor**	**Нативная система**
Безопасность	Доступные системные вызовы	68%	23%	100%
	Векторы атаки через демон	12	3	-
	Интегральная оценка безопасности	0.31	0.74	0.15
Производительность	Вычислительные операции	3...5%	+15...20%	100% (база)
	Операции с файловой системой	+8%	+42%	100%
	Пропускная способность сети	94%	56%	100%
	Сетевая задержка (1 КБ)	1.2 мс	2.8 мс	1.0 мс
Совместимость	Успешные запуски приложений	97%	83%	100%
Проблемные области	Системные вызовы	-	ptrace, ioperm	-
	Сетевые функции	-	Специальные сетевые стеки	-

Сравнение с альтернативными решениями показывает, что gVisor занимает промежуточное положение между традиционной контейнеризацией и полной виртуализацией. По безопасности решение приближается к виртуальным машинам, сохраняя при этом преимущества контейнеров в скорости развертывания и плотности размещения. Однако ограниченная совместимость с некоторыми категориями приложений сужает область практического применения gVisor.

Проведенное исследование позволяет сформулировать четкие критерии выбора технологии изоляции рабочих нагрузок. Docker остается оптимальным решением для производительных систем, обрабатывающих критичные по времени операции, где требования к безопасности могут быть обеспечены дополнительными механизмами на уровне кластера. gVisor следует применять в средах с повышенными требованиями к безопасности, особенно при работе с непроверенным кодом или в мультитенантных окружениях.

Практическая значимость работы подтверждается разработанными рекомендациями по выбору технологии изоляции в зависимости от типа рабочей нагрузки и требований безопасности. Для высокопроизводительных вычислений и сетевых приложений предпочтительнее Docker, в то время как для задач обработки данных и веб-сервисов с умеренными требованиями к производительности может быть выбрана gVisor.

Перспективными направлениями дальнейших исследований являются разработка гибридных решений, сочетающих преимущества обоих подходов, а также создание адаптивных систем изоляции, способных динамически выбирать технологию контейнеризации в зависимости от текущей нагрузки и профиля безопасности. Отдельного внимания заслуживает оптимизация производительности gVisor для специализированных рабочих нагрузок и расширение поддержки системных вызовов для улучшения совместимости.

Список литературы

1. Эдриен Моуэт Использование Docker. - O'Reilly Media, 2017 г.
2. Терских М.Г. Технологии изоляции приложений и инструментальные средства для управления контейнерами // Теория и практика современной науки. - Саратов: Изд-во ООО "Институт управления и социально-экономического развития", 2017 г.

3. Нанян С.М., Ничушкина Т.Н. Виртуальные контейнеры Docker: назначение и особенности применения / Инженерный вестник. - Москва: Изд-во МГТУ им. Н.Э. Баумана, 2015 г. // Теория и практика современной науки.

4. Сейерс Э. Х., Милл А. Docker на практике. - Manning Publishing, 2019 г.

References

1. Edriyen Mouet Ispol'zovaniye Docker. - O'Reilly Media, 2017 g.

2. Terskikh M.G. Tekhnologii izolyatsii prilozheniy i instrumen-tal'nyye sredstva dlya upravleniya konteynerami // Teoriya i praktika sovre-mennoy nauki. - Saratov: Izd-vo OOO "Institut upravleniya i sotsial'no-ekonomicheskogo razvitiya", 2017 g.

3. Nanyan S.M., Nichushkina T.N. Virtual'nyye konteynery Docker: naznacheniye i osobennosti primeneniya / Inzhenernyy vestnik. - Moskva: Izd-vo MGTU im. N.E. Baumana, 2015 g. // Teoriya i praktika sovremennoy nauki.

4. Seyyers E. KH., Mill A. Docker na praktike. - Manning Publishing, 2019 g.