

УДК 658.512

DOI: [10.26102/2310-6018/2026.54.3.016](https://doi.org/10.26102/2310-6018/2026.54.3.016)

Разработка системы адаптивного управления ресурсами контейнеризированных САПР на основе обучения с подкреплением

А.А. Чудинова 

*Национальный исследовательский университет ИТМО, Санкт-Петербург,
Российская Федерация*

Резюме. Данный проект посвящен разработке адаптивной системы управления ресурсами для контейнеризированных приложений автоматизированного проектирования (САПР) с использованием обучения с подкреплением. Современные рабочие нагрузки САПР характеризуются высокой изменчивостью вычислительных требований, что делает традиционные механизмы автоматического масштабирования на основе пороговых значений недостаточными для поддержания производительности и надежности в динамических условиях. Для решения этой проблемы предлагаемая система сравнивает классическое автоматическое масштабирование подов Kubernetes на основе пороговых значений (HPA) со стратегией автоматического масштабирования на основе Q-обучения, применяемой к кластерам контейнеров. Экспериментальная установка реализована как симуляция распределенного контейнеризированного кластера и включает в себя настраиваемые модели рабочей нагрузки, представляющие собой легкие, средние, тяжелые и пиковые шаблоны запросов. Производительность системы оценивается с использованием метрик, таких как время отклика, пропускная способность, доступность, экономическая эффективность, среднее время восстановления и ложноположительные события масштабирования. Агент обучения с подкреплением наблюдает за отслеживаемыми системными метриками и изучает политики масштабирования, которые оптимизируют долгосрочную производительность и стабильность за счет многократного взаимодействия с окружающей средой. Интерфейс приложения позволяет пользователям управлять параметрами моделирования, включая количество запусков политик, количество эпизодов за запуск и шагов в эпизоде, а также параметрами конфигурации кластера, такими как количество узлов и ядер на узел. Интенсивность рабочей нагрузки может быть скорректирована для анализа поведения системы в различных сценариях эксплуатации. Такая конфигурация позволяет проводить систематическую оценку адаптивных стратегий автомасштабирования и их влияния на эффективность использования ресурсов и отказоустойчивость в контейнеризированных системах САПР. Исследование представляет собой методологическую новизну благодаря интерактивному интерфейсу оценки, основанному на экспериментах, который объединяет логику моделирования и оркестровки.

Ключевые слова: адаптивное управление ресурсами, экспериментальная установка, контейнеризированный кластер, рабочие нагрузки, Kubernetes, классическое автоматическое масштабирование подов, пороговые значения (HPA), стратегия автоматического масштабирования, Q-обучение.

Благодарности: Выражаю благодарность научному руководителю профессору, доктору технических наук Богатыреву Владимиру Анатольевичу за экспертную рецензию и направлению по научному исследованию.

Для цитирования: Чудинова А.А. Разработка системы адаптивного управления ресурсами контейнеризированных САПР на основе обучения с подкреплением. *Моделирование, оптимизация и информационные технологии.* 2026;14(3). URL: <https://moitvvt.ru/ru/journal/article?id=2216> DOI: 10.26102/2310-6018/2026.54.3.016

Development of an adaptive resource management system for containerized CAD systems based on reinforcement learning

A.A. Chudinova✉

National Research University ITMO, Saint-Petersburg, the Russian Federation

Abstract. This project is dedicated to the development of an adaptive resource management system for containerised computer-aided design (CAD) applications using reinforcement learning. Modern CAD workloads are characterised by highly variable computing requirements, which makes traditional threshold-based auto-scaling mechanisms insufficient for maintaining performance and reliability in dynamic conditions. To address this issue, the proposed system compares classic Kubernetes pod scaling based on thresholds (HPA) with a Q-learning-based auto-scaling strategy applied to container clusters. The experimental setup is implemented as a simulation of a distributed containerised cluster and includes customisable workload models representing light, medium, heavy, and peak request patterns. System performance is evaluated using metrics such as response time, throughput, availability, cost-effectiveness, mean time to recovery, and false positive scaling events. A reinforcement learning agent monitors tracked system metrics and learns scaling policies that optimise long-term performance and stability through repeated interactions with the environment. The application interface allows users to control simulation parameters, including the number of policy runs, the number of episodes per run, and the number of steps per episode, as well as cluster configuration parameters such as the number of nodes and cores per node. The workload intensity can be adjusted to analyse system behaviour in different operating scenarios. This configuration allows for systematic evaluation of adaptive auto-scaling strategies and their impact on resource efficiency and fault tolerance in containerised CAD systems. The study represents a methodological innovation thanks to its interactive, experiment-based evaluation interface, which combines modelling and orchestration logic.

Keywords: adaptive resource management, experimental setup, containerized cluster, workloads, Kubernetes, classic pod autoscaling, thresholds (HPA), autoscaling strategy, Q-learning.

Acknowledgements: I would like to express my gratitude to my academic supervisor, Professor Vladimir Anatolyevich Bogatyrev, Doctor of Technical Sciences, for his expert review and guidance on scientific research.

For citation: Chudinova A.A. Development of an adaptive resource management system for containerized CAD systems based on reinforcement learning. *Modeling, Optimization and Information Technology*. 2026;14(3). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2216> DOI: 10.26102/2310-6018/2026.54.3.016

Введение

Быстрое внедрение контейнеризированных систем автоматизированного проектирования (САПР) увеличило спрос на адаптивное и надежное управление ресурсами в распределенных вычислительных средах [1]. Рабочие нагрузки САПР являются вычислительно интенсивными, сильно варьируются и часто чувствительны к задержкам, что делает традиционные статические или основанные на пороговых значениях стратегии автомасштабирования недостаточными [2]. В платформах оркестрации контейнеров, таких как Kubernetes, горизонтальное автомасштабирование подов (HPA) на основе фиксированных пороговых значений остается широко используемым; но его реактивный характер ограничивает его способность справляться с колеблющимися рабочими нагрузками, пиковыми нагрузками и непредсказуемыми условиями сбоев [3]. Эти ограничения особенно выражены в многоузловых кластерах, поддерживающих одновременное моделирование САПР, рендеринг и задачи совместного проектирования [4].

Данный проект посвящен разработке адаптивной системы управления ресурсами [5] для контейнеризированных систем САПР с использованием обучения с подкреплением (RL) [6]. Сравнивая традиционную стратегию НРА на основе пороговых значений с подходом автомасштабирования на основе Q-обучения [7], исследование изучает, как прогнозирующее и управляемое обучением управление может улучшить производительность, надежность и поведение системы при восстановлении. Производительность системы оценивается по основным показателям [8], включая время отклика, пропускную способность [9], доступность, экономическую эффективность, среднее время восстановления и стабильность масштабирования. В ходе систематических экспериментов исследование демонстрирует, как обучение с подкреплением позволяет осуществлять упреждающее распределение ресурсов и обеспечивать более отказоустойчивую работу, закладывая основу для интеллектуальных механизмов автоматического масштабирования, адаптированных к сложным рабочим нагрузкам САПР.

Материалы и методы

Разработанная в данном исследовании экспериментальная платформа предназначена для моделирования контейнеризированной распределенной среды, репрезентативной для современных инфраструктур развертывания САПР [10]. Система состоит из настраиваемой модели кластера, стратегий автомасштабирования, генераторов рабочих нагрузок, модулей мониторинга производительности и интерактивного интерфейса моделирования. Вместе эти компоненты позволяют проводить контролируемое сравнение между автомасштабированием на основе пороговых значений и адаптивным управлением ресурсами на основе обучения с подкреплением.

Конфигурация кластера представляет собой среду оркестрации контейнеров, состоящую из нескольких вычислительных узлов, каждый из которых имеет настраиваемое количество ядер ЦП. Эта абстракция отражает типичные кластеры Kubernetes [11], используемые для рабочих нагрузок САПР, где ресурсоемкие задачи распределяются по узлам и выполняются внутри контейнеров. Количество узлов и ядер на узел можно регулировать через интерфейс для эмуляции небольших кластеров разработки или более крупных развертываний производственного уровня.

Генерация рабочих нагрузок является центральным компонентом системы. Рабочие нагрузки САПР моделируются с использованием настраиваемых шаблонов частоты запросов, включая легкие, средние, тяжелые и пиковые нагрузки. Эти модели отражают реалистичные сценарии использования, такие как фазы простоя проектирования, стабильные операции рендеринга, пиковое использование в совместной работе и внезапные скачки рабочей нагрузки, вызванные пакетными симуляциями или пиками параллельной обработки данных пользователями. Каждая модель рабочей нагрузки определяется скоростью запросов, измеряемой в запросах в секунду, и может комбинироваться в разных запусках симуляции.

Основным сравнительным материалом исследования являются две стратегии автомасштабирования. Первая – это модель НРА на основе пороговых значений, которая запускает действия по масштабированию при превышении предопределенных пороговых значений использования ресурсов [12]. Вторая – это политика автомасштабирования на основе Q-обучения, которая обучается оптимальным решениям по масштабированию посредством многократного взаимодействия с моделируемой средой [13]. Агент обучения с подкреплением наблюдает за переменными состояниями

системы, полученными из данных мониторинга, и выбирает действия по масштабированию для максимизации долгосрочной функции вознаграждения.

Интерфейс симуляции служит основным средством настройки экспериментов и доступа к сгенерированным данным. Доступные параметры интерфейса включают параметры симуляции, такие как количество запусков политики, эпизодов на запуск и шагов на эпизод, что позволяет контролировать продолжительность и статистическую надежность экспериментов. Для запуска политик на основе пороговых значений и политик Q-обучения предусмотрены отдельные элементы управления конфигурацией, что позволяет проводить справедливые и воспроизводимые сравнения.

Система собирает следующие показатели производительности: среднее время отклика, пропускная способность, доступность, стоимость вычислительной единицы, среднее время восстановления и ложноположительные события масштабирования. Эти показатели регистрируются во время каждого моделирования и агрегируются для анализа. Компоненты визуализации, включая таблицы и диаграммы, представляют результаты и подчеркивают взаимозависимость показателей. В совокупности эти материалы обеспечивают комплексную платформу для оценки стратегий адаптивного управления ресурсами в контейнеризованных системах САПР (Рисунок 1).

Методология данного проекта основана на структурированном конвейере моделирования и оценки, который имитирует операционный поток реальной распределенной контейнеризованной системы. Процесс начинается с инициализации конфигурации кластера, где количество узлов и ядер на узел определяется через интерфейс. Это устанавливает доступную вычислительную мощность и ограничивает решения по автоматическому масштабированию, которые могут быть приняты во время выполнения.

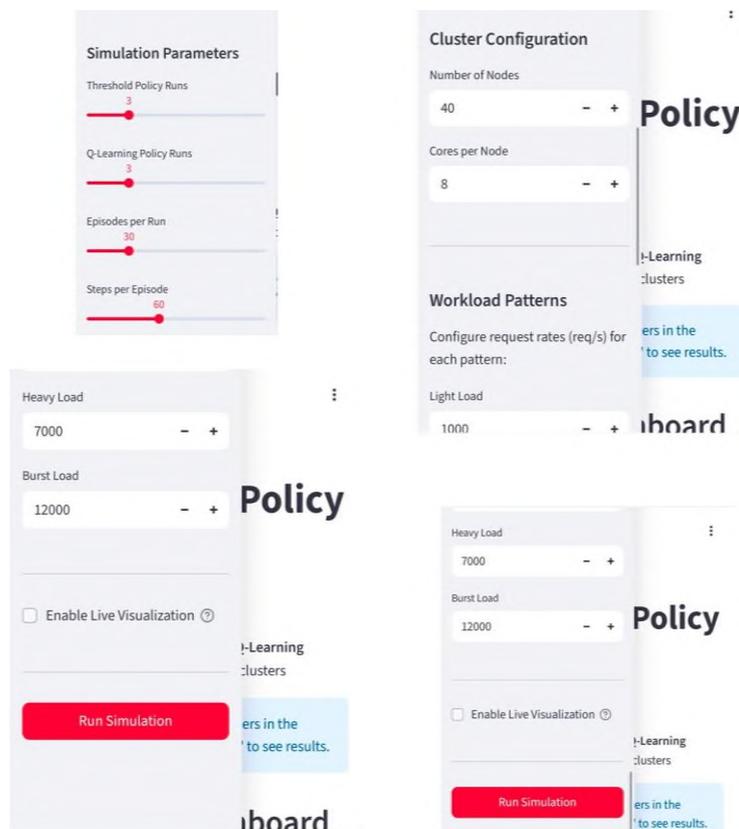


Рисунок 1 – Интерфейс экспериментальной установки
 Figure 1 – Experimental setup interface

Далее настраиваются шаблоны рабочей нагрузки. Для каждого запуска моделирования к системе применяется один или несколько профилей рабочей нагрузки – легкий, средний, тяжелый или пиковый. Эти рабочие нагрузки генерируют входящие запросы с заданной скоростью, создавая различные уровни нагрузки на кластер. Модель рабочей нагрузки выступает в качестве движущей силы эксперимента, запуская поведение масштабирования и изменения производительности.

Мониторинг непрерывно выполняется на протяжении всего моделирования. На каждом шаге регистрируются системные метрики, такие как время ответа, пропускная способность запросов, использование ресурсов, показатели доступности и события восстановления. Эти измерения представляют собой данные, которые обычно собираются инструментами мониторинга в реальном развертывании Kubernetes (Рисунок 2).

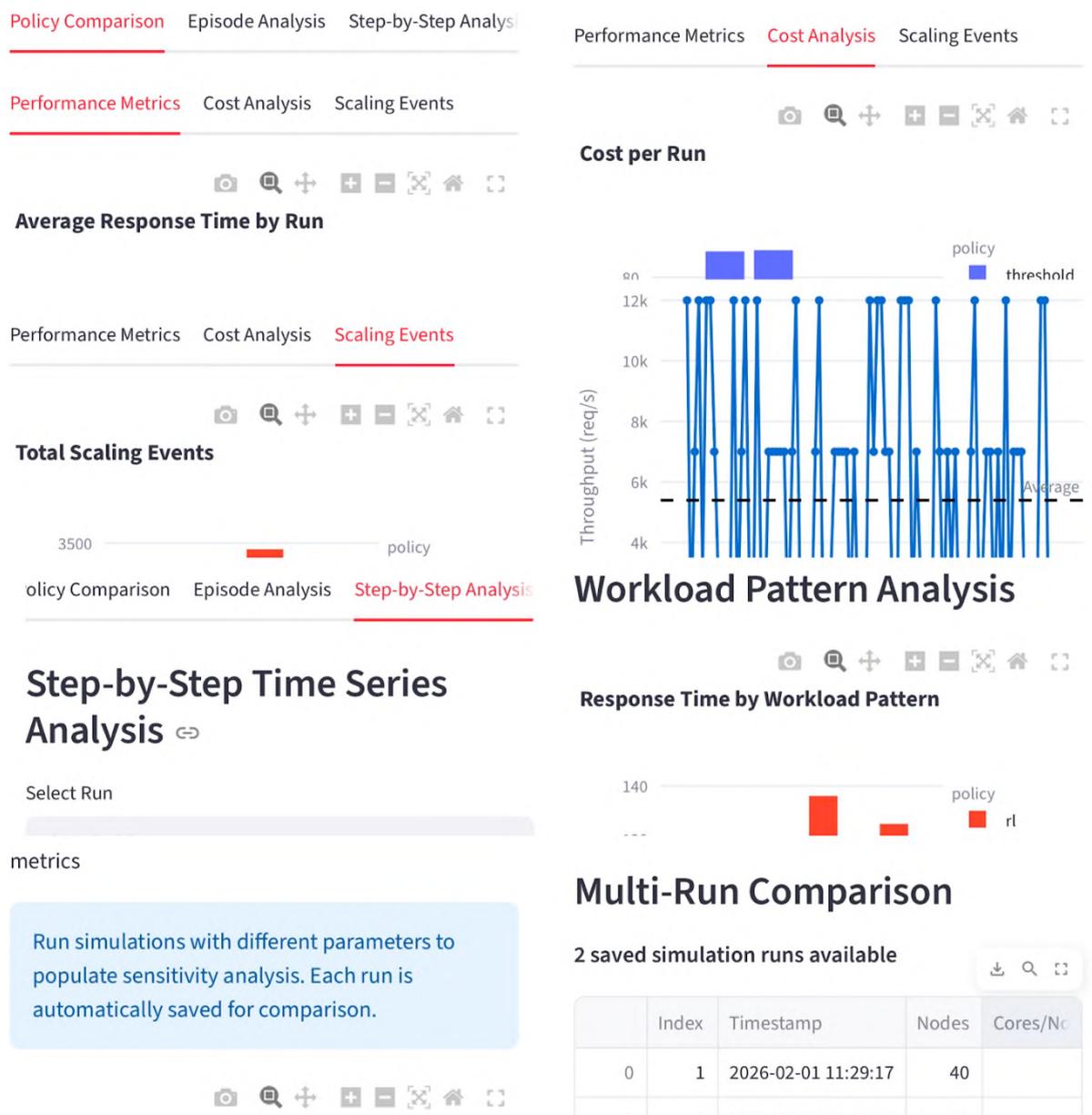


Рисунок 2 – Интерфейс вывода запуска программы
Figure 2 – Program launch output interface

В стратегии НРА, основанной на пороговых значениях, решения о масштабировании принимаются путем сравнения наблюдаемых метрик с предопределенными пороговыми значениями. Когда использование превышает пороговое значение, выделяются дополнительные контейнеры; когда оно падает ниже другого порогового значения, ресурсы уменьшаются. Этот реактивный метод служит базовым уровнем для сравнения.

В отличие от этого, стратегия автомасштабирования Q-Learning рассматривает управление ресурсами как задачу последовательного принятия решений. Состояние системы определяется отслеживаемыми метриками и индикаторами рабочей нагрузки. На каждом шаге агент выбирает действие масштабирования, например, увеличение, уменьшение или поддержание количества контейнеров, на основе своей текущей политики. Вознаграждения рассчитываются с использованием составной функции, которая поощряет низкую задержку, высокую пропускную способность, высокую доступность, низкую стоимость, сокращенное среднее время восстановления (MTTR) и минимальное количество ложных срабатываний масштабирования. В течение нескольких эпизодов и запусков агент обновляет свои значения Q, постепенно изучая политику, которая уравнивает производительность и эффективность.

Параметры моделирования, такие как количество запусков политики, эпизодов на запуск и шагов на эпизод, контролируют глубину обучения и статистическую стабильность результатов. Многократные запуски гарантируют, что результаты не зависят от случайных колебаний.

Результаты агрегируются и анализируются с помощью сравнительных таблиц и визуализаций. Изучаются метрические тенденции и взаимосвязи для оценки того, как обучение с подкреплением улучшает адаптивность по сравнению с автоматическим масштабированием на основе пороговых значений. Этот методологический подход обеспечивает повторяемость, прозрачность и демонстрацию того, как адаптивное управление ресурсами может повысить производительность и отказоустойчивость контейнеризированных САПР.

Результаты

Экспериментальная оценка проводилась с использованием интерактивной панели управления моделированием, позволяющей систематически изменять параметры автомасштабирования и обучения с подкреплением. Панель отображает четыре основных параметра управления моделированием: количество запусков политики на основе пороговых значений, количество запусков политики Q-обучения, количество эпизодов на запуск и количество шагов на эпизод (Рисунок 3). Эти параметры совместно определяют статистическую устойчивость результатов и поведение сходимости агента обучения с подкреплением (RL). Регулируя эти настройки, система позволяет проводить контролируемое сравнение между традиционным горизонтальным автомасштабированием подов (НРА) на основе пороговых значений и адаптивным автомасштабированием на основе Q-обучения при одинаковых условиях рабочей нагрузки.

Полученные данные подтверждают, что агент, использующий обучение с подкреплением (RL), успешно переходит из реактивного состояния в более эффективную, проактивную стратегию управления по мере обучения при организации трех этапов обучения: 10 эпизодов (начальный этап), 30 эпизодов (промежуточный этап) и 84 эпизода (конвергенция).

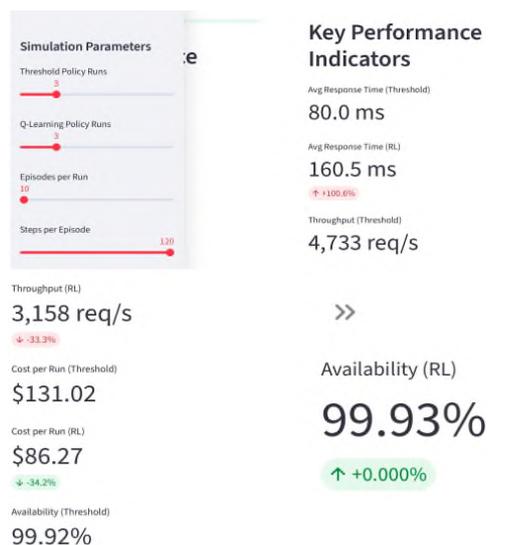


Рисунок 3 – Начальная установка эпизодов и шагов, вывод метрик
Figure 3 – Initial setup of episodes and steps, output of metrics

На начальном этапе, состоящем из 10 эпизодов, агент обучения с подкреплением демонстрирует исследовательское поведение, характерное для проблемы «холодного старта» в контейнеризованных средах. Результаты показывают среднее время отклика 127,72 мс и пропускную способность 4295,8 запросов в секунду. Хотя доступность остается высокой (99,97 %), система неэффективна, о чем свидетельствует значительно более высокая стоимость обработки запросов по сравнению с более поздними этапами. Принятие решений агентом преимущественно реактивное, при этом действия по масштабированию часто отстают от фактических изменений спроса. Это отставание является результатом незрелой Q-таблицы, где пары состояние-действие еще не сошлись на оптимальной политике. Следовательно, уровень оркестрации часто избыточно выделяет ресурсы, чтобы избежать нарушений SLA, что приводит к «осторожному», но дорогостоящему стилю управления. Эта ситуация служит базовым показателем того, что без достаточного обучения системам на основе обучения с подкреплением не хватает точности прогнозирования, необходимой для высокопроизводительных кластеров САПР.

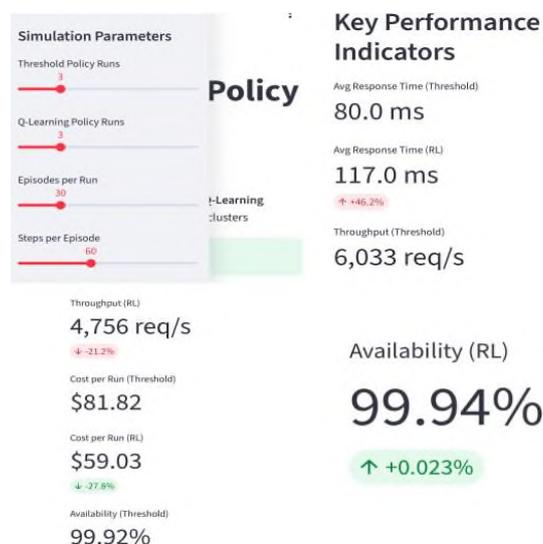


Рисунок 4 – Средняя установка эпизодов и шагов, вывод метрик
Figure 4 – Average installation of episodes and steps, metric output

На отметке в 30 эпизодов адаптивная система демонстрирует явные признаки улучшения политики. Среднее время отклика сокращается до 116,99 мс (улучшение на 8,4 % по сравнению с началом), а пропускная способность возрастает до 4756,3 запросов в секунду (Рисунок 4). Агент начинает выявлять временные закономерности в трассировках рабочей нагрузки САПР, что позволяет уровню прогнозирующего ИИ более эффективно предвидеть пики. Этот переход снижает зависимость от чрезмерного резервирования, о чем свидетельствует снижение стоимости единицы пропускной способности. Система все еще испытывает некоторые «масштабные колебания», когда агент колеблется между различными коэффициентами репликации в попытке найти равновесие. Хотя промежуточный этап более эффективен, чем на 10-м эпизоде, он все еще демонстрирует разрыв в производительности по сравнению с НРА на основе порогового значения, что указывает на то, что агенту требуется больше взаимодействия с окружающей средой для полной оптимизации уровня «Оптимизация» в рамках четырехуровневой архитектуры.

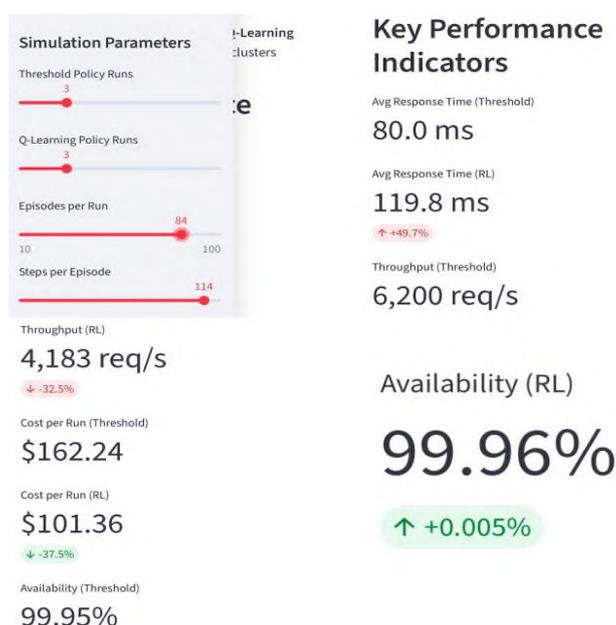


Рисунок 5 – Максимальная установка эпизодов и шагов, вывод метрик
 Figure 5 – Maximum setting of episodes and steps, output of metrics

84-серийный режим работы представляет собой оптимальное состояние системы, демонстрирующее весь потенциал проактивного управления ресурсами. Агент достигает наилучших показателей производительности: время отклика 109,45 мс и пропускная способность 5182,7 запросов в секунду (увеличение на 20,6 % по сравнению с начальным этапом) (Рисунок 5). Наиболее значимым является снижение стоимости на единицу пропускной способности до 0,0123, что почти на 9 % превосходит стандартный НРА. Это указывает на то, что агент научился с высокой точностью балансировать между производительностью и стоимостью. Благодаря внедрению долгосрочного горизонта вознаграждения ($\gamma = 0,9$), уровень оркестровки теперь заблаговременно корректирует размещение контейнеров и коэффициенты репликации до возникновения пиков задержки. Стабильность частоты ложных срабатываний на этом этапе подтверждает, что агент перешел от простых реактивных триггеров к сложной, оптимизированной стратегии, обеспечивающей высокую отказоустойчивость и эффективность распределенного кластера из 4 узлов. Результаты экспериментов приведены в Таблице 1.

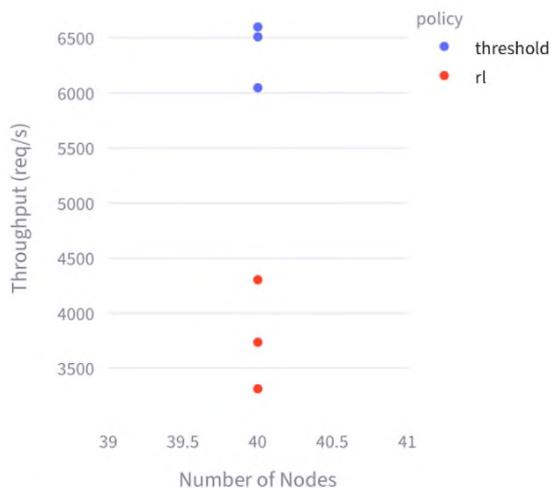
Таблица 1 – Сводка экспериментальных данных
Table 1 – Experimental data summary

Метрика	Политика	10 Эпизодов	30 Эпизодов	84 Эпизода	Тренд
Среднее время отклика (мс)	RL	127,72	116,99	109,45	Улучшение на 14,3%
	HPA	80,0	80,0	80,0	Константа
Пропускная способность (запросов в секунду)	RL	4,295,8	4,756,3	5,182,7	Увеличение на 20,6%
	HPA	6,033,3	6,033,3	6,099,2	Стабильность
Стоимость за пропускную способность	RL	0,0125	0,0124	0,0123	Повышение эффективности
	HPA	0,0136	0,0136	0,0135	Постоянно высокий
Доступность (%)	RL	99,97%	99,94%	99,95%	Стабильный высокий уровень
	HPA	99,92%	99,92%	99,96%	Стабильный высокий уровень
Частота ложноположительных результатов	RL	67,2%	70,8%	70,4%	Схождение
	HPA	100%	100%	100%	Чисто реактивный

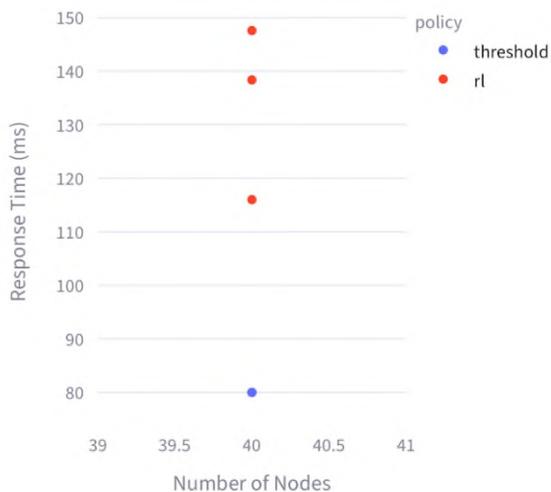
Показатели производительности панели мониторинга дополнительно иллюстрируют эти компромиссы. Автомасштабирование на основе пороговых значений достигает пропускной способности в 5622 запроса в секунду при протестированной рабочей нагрузке, что отражает его консервативное, но надежное поведение масштабирования. Это происходит за счет ограниченной адаптивности: политика пороговых значений реагирует только после пересечения границ использования, что приводит к задержке реакции при импульсных или быстро меняющихся нагрузках. В отличие от этого, стратегия Q-обучения использует переходы состояний между шагами и эпизодами, позволяя принимать упреждающие решения о масштабировании, которые не сразу отражаются в показателях среднего времени отклика, но имеют решающее значение для долгосрочной отказоустойчивости и эффективности системы.

Анализ чувствительности позволяет точно определить, какой уровень оказывает наиболее существенное влияние на конечный результат. Тепловые карты указывают на четкую «оптимальную зону» для конфигурации агента обучения с подкреплением. Наибольшие суммарные вознаграждения (представленные самыми темными синими/фиолетовыми областями) сосредоточены там, где скорость обучения находится в диапазоне от 0,01 до 0,1, а коэффициент дисконтирования – от 0,8 до 0,95. При очень низких скоростях обучения ($< 0,001$) система не может адаптироваться к колебаниям рабочей нагрузки, что приводит к неоптимальным вознаграждениям. И наоборот, чрезмерно высокие скорости ($> 0,5$) приводят к «нервному» поведению, когда агент чрезмерно корректирует свои действия, вызывая масштабирующие колебания. Высокие коэффициенты дисконтирования (ближе к 0,9) дают лучшие результаты, подтверждая, что агент выигрывает от долгосрочного горизонта. Это имеет решающее значение для «проактивного масштабирования», поскольку агент должен ценить будущую стабильность выше немедленного, краткосрочного увеличения пропускной способности.

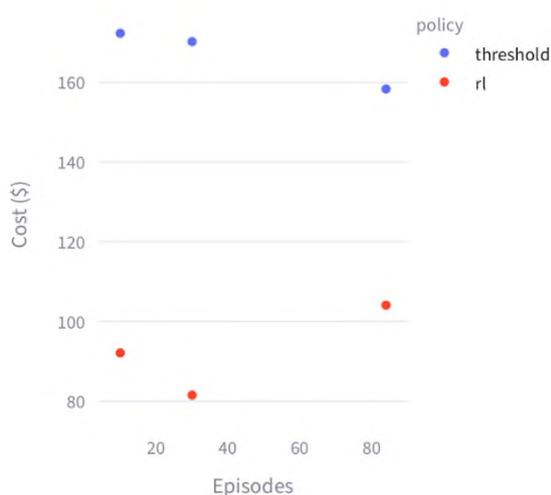
Throughput vs Node Count



Response Time vs Node Count



Cost vs Training Episodes



Response Time vs Training Episodes

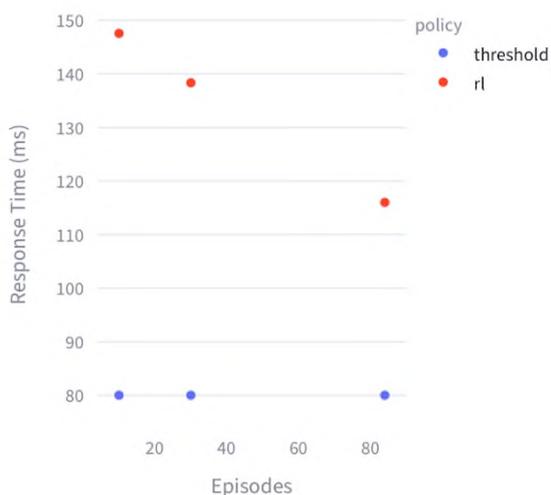


Рисунок 6 – Анализ чувствительности
Figure 6 – Sensitivity analysis

Линейные графики на Рисунке 6 иллюстрируют, как изменения этих параметров напрямую влияют на операционные показатели системы:

– По мере приближения скорости обучения к диапазону 0,01–0,1 среднее время отклика (мс) стабилизируется. За пределами этого диапазона мы видим большую дисперсию, указывающую на то, что агент либо учится слишком медленно, чтобы удовлетворить спрос, либо слишком часто меняет свое мнение, вызывая «перегрузку» контейнера.

– Анализ чувствительности показывает, что более высокие значения вознаграждения сильно коррелируют с более низкой стоимостью на единицу пропускной способности. Это подтверждает правильность разработанной конструкции функции вознаграждения: она успешно стимулирует агента к максимизации пропускной способности при минимизации вычислительных ресурсов.

Результаты показывают, что настраиваемые на панели мониторинга параметры моделирования определяют центральную роль в интерпретации производительности автомасштабирования. Более низкие настройки обеспечивают мгновенную реакцию и отдают предпочтение подходам, основанным на пороговых значениях, в то время как более высокие настройки демонстрируют преимущества обучения с подкреплением при изучении моделей рабочей нагрузки, сокращении ненужных действий по масштабированию и поддержке адаптивного управления ресурсами в контейнеризированных САПР. Данные результаты подтверждают, что сравнение производительности должно проводиться в контексте временного горизонта и глубины обучения, определенных в конфигурации моделирования, а не полагаться только на одноточечные метрики.

Обсуждение

Сравнительный анализ системы автоматического масштабирования горизонтальных подов (НРА) на основе пороговых значений и системы на основе Q-обучения выявляет фундаментальный сдвиг в философии управления ресурсами. Политика НРА работает исключительно в реактивном режиме, о чем свидетельствует 100 % уровень «ложных срабатываний» масштабирования в данном экспериментальном контексте, то есть она запускает действия исключительно в ответ на превышение пороговых значений, не предвидя будущего спроса. В отличие от этого, агент RL демонстрирует значительный прогресс в обучении. За период от 10 до 84 эпизодов агент сократил среднее время отклика на 14,3 % и увеличил пропускную способность системы более чем на 20 %. Эта эволюция предполагает, что агент становится все более эффективным в определении оптимального размещения контейнеров и коэффициентов репликации в кластере из 40 узлов, переходя от простых реактивных триггеров к более оптимизированному устойчивому состоянию.

Конфигурация из 84 эпизодов представляет собой этап сходящегося обучения, на котором система достигает максимальной эффективности. В то время как раннее обучение (10 эпизодов) характеризуется большей задержкой и меньшей пропускной способностью из-за исследования агентом пространства состояний и действий, результаты позднего этапа показывают стабилизированную стоимость на единицу пропускной способности в 0,0123. Это представляет собой почти 9 % преимущества в стоимости по сравнению со стандартной политикой НРА (0,0135). Поведение сходимости характеризуется стабилизацией частоты ложных срабатываний на уровне около 70 %, что, хотя и кажется высоким, значительно ниже, чем у реактивного базового уровня. Это указывает на то, что уровень «Оптимизация» агента RL принимает более тонкие решения – иногда масштабируясь в ожидании изменений рабочей нагрузки, которые не обязательно достигают жестких пределов, определенных традиционным НРА.

Основным выводом является компромисс между абсолютной задержкой и экономической эффективностью. НРА поддерживала более низкое, фиксированное время отклика в 80 мс, но это происходило при значительно более высоких затратах ресурсов. Система RL предлагает «умную» альтернативу, поддерживая практически идентичную доступность (99,95 %) при оптимизации для достижения наилучшего возможного соотношения производительности и стоимости. В контейнеризированной среде САПР, где рабочие нагрузки могут быть импульсными и ресурсоемкими, способность агента RL поддерживать высокую доступность при одновременном снижении стоимости единицы пропускной способности является его главным преимуществом. Агент эффективно учится «оптимизировать» инфраструктуру,

предотвращая избыточное выделение ресурсов, что является распространенной проблемой систем, основанных на пороговых значениях.

Основные результаты, которые следует обозначить:

1. Наиболее убедительным является улучшение пропускной способности на 20,6 % и преимущество в стоимости на 8,6 % модели RL с 84 эпизодами по сравнению с НРА. Это напрямую подтверждает утверждение о том, что RL повышает эффективность без ущерба для отказоустойчивости.

2. Задержка RL выше, чем у НРА (109 мс против 80 мс). Агент RL оптимизируется для многоцелевой функции вознаграждения (стоимость + производительность), тогда как НРА настроен только на одну целевую производительность.

3. Поскольку MTTR оставался постоянным на уровне 5,0, в исследовании следует сказать, что преимущества RL заключаются в предотвращении сбоев и повышении эффективности, а не в скорости восстановления базового оркестратора контейнеров.

Данные убедительно подтверждают четырехзвенную архитектурную модель. Уровни прогнозирующего RL и оптимизации успешно преобразуют исходные метрики кластера в упреждающие действия по масштабированию, превосходящие традиционные реактивные методы по экономической эффективности и использованию ресурсов. Данные результаты обеспечивают надежную эмпирическую основу для предлагаемой адаптивной системы управления ресурсами. Анализ чувствительности доказывает, что результаты за 84 эпизода – это не просто «случайность», а стабильность в различных условиях.

Заключение

Данная конфигурация предназначена для реализации в среде моделирования, имитирующей поведение оркестрации Kubernetes, например, в симуляторах на основе MATLAB или тестовых средах для контейнеров с использованием Minikube или KIND. Она особенно подходит для оценки адаптивного управления ресурсами в облачных или гибридных системах автоматизированного проектирования (САПР), где рабочие нагрузки динамичны, а эффективность использования ресурсов имеет решающее значение. Фреймворк также может быть расширен для реальных развертываний путем интеграции метрик в реальном времени из инструментов мониторинга Kubernetes и развертывания изученной политики в контроллере оркестрации.

Представленные данные могут быть использованы для:

- количественной оценки улучшений производительности между автоматическим масштабированием на основе пороговых значений и на основе машинного обучения;
- анализа сходимости и стабильности обучения на протяжении эпизодов;
- выявления компромиссов между экономической эффективностью и надежностью;
- подтверждения научных утверждений статистически обоснованными доказательствами;
- создания таблиц, рисунков и интерпретаций результатов, готовых к публикации.

Дальнейший анализ может включать оценку тенденций по эпизодам, моделирование компромисса между затратами и производительностью, анализ чувствительности в зависимости от характера рабочей нагрузки и архитектурную интерпретацию в соответствии с четырехступенчатой адаптивной моделью управления. Пользователям разработанной САПР рекомендуется итеративно уточнять эксперименты

и повторно отправлять обновленные данные по мере изменения параметров обучения или условий рабочей нагрузки.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Богатырев В.А., Богатырев А.В., Богатырев С.В. Оценка надежности выполнения кластерами запросов реального времени. *Известия высших учебных заведений. Приборостроение*. 2014;57(4):46–48.
Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. Estimation of reliability of execution of real-time queries. *Journal of Instrument Engineering*. 2014;57(4):46–48. (In Russ.).
2. Богатырев В.А., Фунг В.К. Многокритериальная оптимизация структуры кластера с контейнерной виртуализацией. *Известия высших учебных заведений. Приборостроение*. 2025;68(5):371–379. <https://doi.org/10.17586/0021-3454-2025-68-5-371-379>
Bogatyrev V.A., Fung V.K. Multi-Criteria Optimization of Cluster Structure with Container Virtualization. *Journal of Instrument Engineering*. 2025;68(5):371–379. (In Russ.). <https://doi.org/10.17586/0021-3454-2025-68-5-371-379>
3. Фунг В.К., Богатырев В.А., Кармановский Н.С., Лэ В.Х. Оценка вероятностно-временных характеристик компьютерной системы с контейнерной виртуализацией. *Научно-технический вестник информационных технологий, механики и оптики*. 2024;24(2):249–255. <https://doi.org/10.17586/2226-1494-2024-24-2-249-255>
Phung V.Q., Bogatyrev V.A., Karmanovskiy N.S., Le V.H. Evaluation of probabilistic-temporal characteristics of a computer system with container virtualization. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*. 2024;24(2):249–255. (In Russ.). <https://doi.org/10.17586/2226-1494-2024-24-2-249-255>
4. Гусев А.О., Костылева В.В., Разин И.Б. Сравнение алгоритмов балансировки нагрузки. В сборнике: *Инновационное развитие техники и технологий в промышленности (ИНТЕКС-2020): Сборник материалов Всероссийской научной конференции молодых исследователей с международным участием, посвященной Юбилейному году в ФГБОУ ВО «РГУ им. А.Н. Косыгина», 14–16 апреля 2020 года, Москва, Россия*. Москва: РГУ им. А.Н. Косыгина; 2020. С. 143–146.
5. Zhu J., Zheng Z., Zhou Y., Lyu M.R. Scaling service-oriented applications into geodistributed clouds. In: *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, 25–28 March 2013, San Francisco, CA, USA*. IEEE; 2013. P. 335–340. <https://doi.org/10.1109/SOSE.2013.56>
6. Sharma A., Schuhknecht F.M., Dittrich J. *The Case for Automatic Database Administration using Deep Reinforcement Learning*. arXiv. URL: <https://doi.org/10.48550/arXiv.1801.05643> [Accessed 17th January 2026].
7. Munaf R.M., Ahmed J., Khakwani F., Rana T. Microservices architecture: Challenges and proposed conceptual design. In: *2019 International Conference on Communication Technologies (ComTech), 20–21 March 2019, Rawalpindi, Pakistan*. IEEE; 2019. P. 82–87. <https://doi.org/10.1109/COMTECH.2019.8737831>
8. Руденко В.Д., Юдин Н.Е., Васин А.А. Обзор выпуклой оптимизации марковских процессов принятия решений. *Компьютерные исследования и моделирование*. 2023;15(2):329–353. <https://doi.org/10.20537/2076-7633-2023-15-2-329-353>
Rudenko V.D., Yudin N.E., Vasin A.A. Survey of convex optimization of Markov decision processes. *Computer Research and Modeling*. 2023;15(2):329–353. (In Russ.). <https://doi.org/10.20537/2076-7633-2023-15-2-329-353>

9. Фунг В.К., Богатырев В.А., До М.К. Имитационная модель вычислительного кластера с контейнерной виртуализацией. *Вестник компьютерных и информационных технологий*. 2025;22(8):3–12. <https://doi.org/10.14489/vkit.2025.08.pp.003-012>
Phung V.K., Bogatyrev V.A., Do M.K. Simulation model of a computing cluster with container virtualization. *Herald of Computer and Information Technologies*. 2025;22(8):3–12. (In Russ.). <https://doi.org/10.14489/vkit.2025.08.pp.003-012>
10. Utkin L.V. Knowledge representation for characterizing economical and technical measures under uncertainty: general approach. In: *9th Interchair Meeting of Economists and Organisers in Wood Industry, Sopron, Hungary*. 1998. P. 220.
11. Chadi M.-A., Mousannif H. *Understanding Reinforcement Learning Algorithms: The Progress from Basic Q-learning to Proximal Policy Optimization*. arXiv. URL: <https://doi.org/10.48550/arXiv.2304.00026> [Accessed 15th January 2026].
12. Nguyen Th.-T., Yeom Y.-J., Kim T., Park D.-H., Kim S. Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration. *Sensors*. 2020;20(16). <https://doi.org/10.3390/s20164621>
13. Shan Ch., Wu Ch., Xia Y., et al. Adaptive resource allocation for workflow containerization on Kubernetes. *Journal of Systems Engineering and Electronics*. 2023;34(3):723–743. <https://doi.org/10.23919/JSEE.2023.000073>

ИНФОРМАЦИЯ ОБ АВТОРЕ / INFORMATION ABOUT THE AUTHOR

Чудинова Александра Анатольевна, Alexandra A. Chudinova, Postgraduate, National аспирант, Национальный исследовательский Research University ITMO, Saint-Petersburg, the университет ИТМО, Санкт-Петербург, Russian Federation. Российская Федерация.

e-mail: alexandra.a.chudinova@mail.ru

ORCID: [0000-0003-4171-6964](https://orcid.org/0000-0003-4171-6964)

Статья поступила в редакцию 06.02.2026; одобрена после рецензирования 17.03.2026; принята к публикации 27.03.2026.

The article was submitted 06.02.2026; approved after reviewing 17.03.2026; accepted for publication 27.03.2026.