

УДК 004.054

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ СЕРВИСА YANDEX LOAD TESTING ДЛЯ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ ВЕБ-СЕРВИСОВ



К.А. Кошлатый

Студент третьего курса
факультета компьютерного
проектирования БГУИР
alfnish159@gmail.com@gmail.com



И.В. Лихтар

Студент третьего курса
факультета компьютерного
проектирования БГУИР
llikhtarilya141112005@gmail.com



А.Ю. Ефремова

Ассистент кафедры
проектирования
информационно-
компьютерных систем
БГУИР
al617e13@gmail.com

К.А. Кошлатый

Обучается в Белорусском университете информатики и радиоэлектроники. Область научных интересов связана с анализом данных и методами тестирования.

И.В. Лихтар

Обучается в Белорусском университете информатики и радиоэлектроники. Область научных интересов связана с исследованием методологий разработки.

А.Ю. Ефремова

Окончила Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с обеспечением информационной безопасности и надежного функционирования цифровых систем.

Аннотация. В статье рассматривается применение Yandex cloud testing как эффективного инструмента для оценки производительности веб-сервисов.

Ключевые слова: тестирование, тест-кейсы, оптимизации проектных решений, облачное тестирование.

Введение. В условиях современного мира и гонки технологий все большее количество пользовательских и бизнес-процессов переносится в веб среду. Веб-сервисы используются повсеместно: магазины, поликлиники, аптеки, военная индустрия. При высокой нагрузке из-за множества подключений и больших объемов данных от сервиса требуется стабильность работы и приемлемое время отклика.

Объектом тестирования является веб-сервис для аналитики и поддержания бизнес-процессов тату салона. Сервис построен на архитектуре микросервисов с архитектурным стилем RESTful API.

Целью работы является изучение теоретических основ нагрузочного тестирования и анализ возможностей современных облачных средств, применяемых для проверки производительности веб-сервисов.

Теоретические основы тестирования. Нагрузочное тестирование рассматривается как часть более обширного тестирования производительности, при котором система находится под ожидаемой и/или целевой нагрузкой. При этом оцениваются ее устойчивость и стабильность: выдержит ли сервис высокую нагрузку и большой наплыв данных или

пользователей. В современных условиях все большую популярность приобретают облачные сервисы тестирования. Их применения позволяет сократить время тестирования, что позволяет снизить временные и материальные затраты. Они дают возможность запускать тесты без развертывания сложной локальной архитектуры, быстро масштабировать объемы нагрузки и получать результаты в простой для понимания форме.

Тестирование производительности помогает собрать измеримые данные о рабочей нагрузке. Ранний запуск тестов помогает создавать рабочие нагрузки в соответствии с правильными спецификациями [1].

Нагрузочное тестирование является одним из ключевых видов тестирования производительности и применяется для оценки поведения программной системы при ожидаемой рабочей нагрузке. Его основная задача состоит в том, чтобы определить, способна ли система стабильно функционировать при нормальном и близком к пиковому количестве пользователей, запросов или операций.

Облачное нагрузочное тестирование представляет собой подход, при котором генерация нагрузки, управление тестовыми сценариями и сбор данных происходят с использованием облачной инфраструктуры.

Одной из главных особенностей облачного нагрузочного тестирования является масштабируемость. При использовании облачной платформы объем генерируемой нагрузки может увеличиваться значительно быстрее, чем при локальном подходе, поскольку пользователю не требуется самостоятельно подготавливать дополнительные технические средства, настраивать распределение нагрузки и поддерживать работоспособность всей тестовой среды.

Еще одной особенностью является централизованный сбор и визуализация результатов. Облачные решения, как правило, предоставляют отчеты, графики и панели мониторинга, в которых отражаются ключевые показатели теста: время отклика, число успешных и ошибочных запросов, интенсивность нагрузки и поведение системы во времени.

Отдельно следует отметить, что облачный подход особенно удобен в учебных, исследовательских и прикладных задачах, где важно быстро получить результат без значительных затрат на поддержку собственной тестовой инфраструктуры.

Можно выделить несколько требований к средам нагрузочного тестирования:

- масштабируемость;
- система аналитики и мониторинга;
- возможность интеграции с CI/CD [2]

Именно совокупность этих критериев и показывает, насколько инструмент пригоден для объективной оценки сервиса.

Yandex Load Testing представляет собой облачный сервис, предназначенный для проведения нагрузочного тестирования с возможностью создания настроенного генератора нагрузки в облаке, запуска тестов через интерфейс, просмотра отчетов и хранения результатов предыдущих запусков для последующего сравнения. Практическая ценность сервиса заключается в том, что он объединяет в одной среде подготовку тестовой инфраструктуры, выполнение нагрузочного сценария и анализ полученных результатов. Пользователь получает возможность не только сгенерировать нужный поток запросов к тестируемому веб-сервису, но и отслеживать поведение системы по ключевым метрикам, включая время отклика, квантили, коды ответов и динамику нагрузки.

Архитектура Yandex cloud testing состоит из нескольких основных компонентов: тестируемый сервис, агент, конфигурация теста и payload.

Тестируемый сервис является целевым объектом, на который направляется нагрузка. Это может быть веб-приложение, API, HTTP(S)-сервис или иная система, доступная агенту [3].

Агент же является виртуальной машиной, на которой размещены инструмент нагрузочного тестирования, генераторы нагрузки и коннектор для взаимодействия с сервисом Yandex Load Testing.

Сервис был выбран ввиду ряда преимуществ, описанных далее.

Основное преимущество – облачный характер сервиса, позволяющий выполнять работу гораздо более эффективно и нетребовательно к комплектующим технического средства, на котором мы работаем.

Второе – встроенная аналитика, благодаря которой сервис хорошо подходит для анализа производительности backend-компонентов, но не заменяет инструменты UI/E2E-тестирования.

Третье – поддержка разных генераторов и сценариев. Пользователь может применять Pandora, Phantom и, по сервисной странице, JMeter, выбирать формат тестовых данных и использовать файлы из Object Storage.

Проведение исследования. Перед проведением нагрузочного тестирования необходимо подготовить тестовое окружение в Yandex Cloud. Для работы с Yandex Load Testing требуется тестируемый сервис, агент генерации нагрузки, конфигурация теста и тестовые данные. В рамках исследования в качестве тестируемого приложения рассматривается информационная система тату-салона с аналитическим модулем, а генерация нагрузки должна выполняться с помощью отдельного агента Load Testing.

Подготовка окружения включает создание сервисного аккаунта для работы агента, настройку сети и создание виртуальной машины агента в сервисе Load Testing. В документации Yandex Cloud для агента указывается сервисный аккаунт, а также подсеть с настроенным NAT-шлюзом; для взаимодействия агента с сервисом требуется исходящий доступ по порту 443 (рисунок 1).

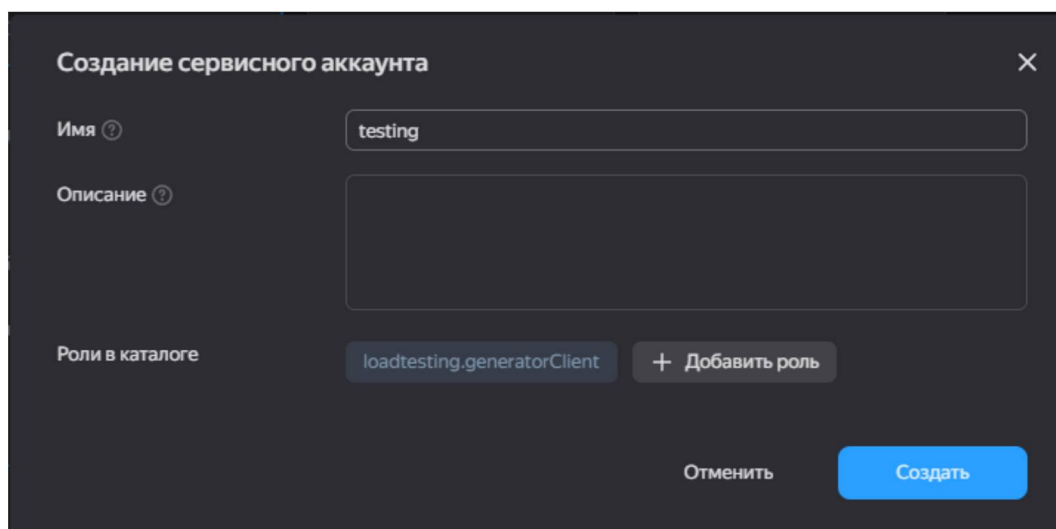


Рисунок 1. Интерфейс создания сервисного аккаунта

При подготовке окружения также важно учитывать параметры самого агента, поскольку от его ресурсов зависит корректность генерации нагрузки. Мониторинг состояния агента может быть использован для контроля загрузки процессора, памяти, сети и других параметров во время тестирования.

Yandex Cloud предоставляет несколько типов агентов, отличающихся вычислительными характеристиками: small, medium и Large. Каждый отличается по количеству памяти и вычислительным способностям. Small обладает наименьшим количеством оперативной памяти и объемом хранилища.

Он подходит для небольших обучающих проектов. Medium уже более продвинутый агент, с ним можно вести тестирование небольших коммерческих проектов. Large уже обладает внушительными объемами памяти, с которым можно смело тестировать большие коммерческие проекты со сложной структурой и большим количеством внутренних процессов.

Это необходимо для того, чтобы источник нагрузки сам не стал ограничивающим фактором эксперимента.

В контексте исследования выбран агент со средними характеристиками (рисунок 2).

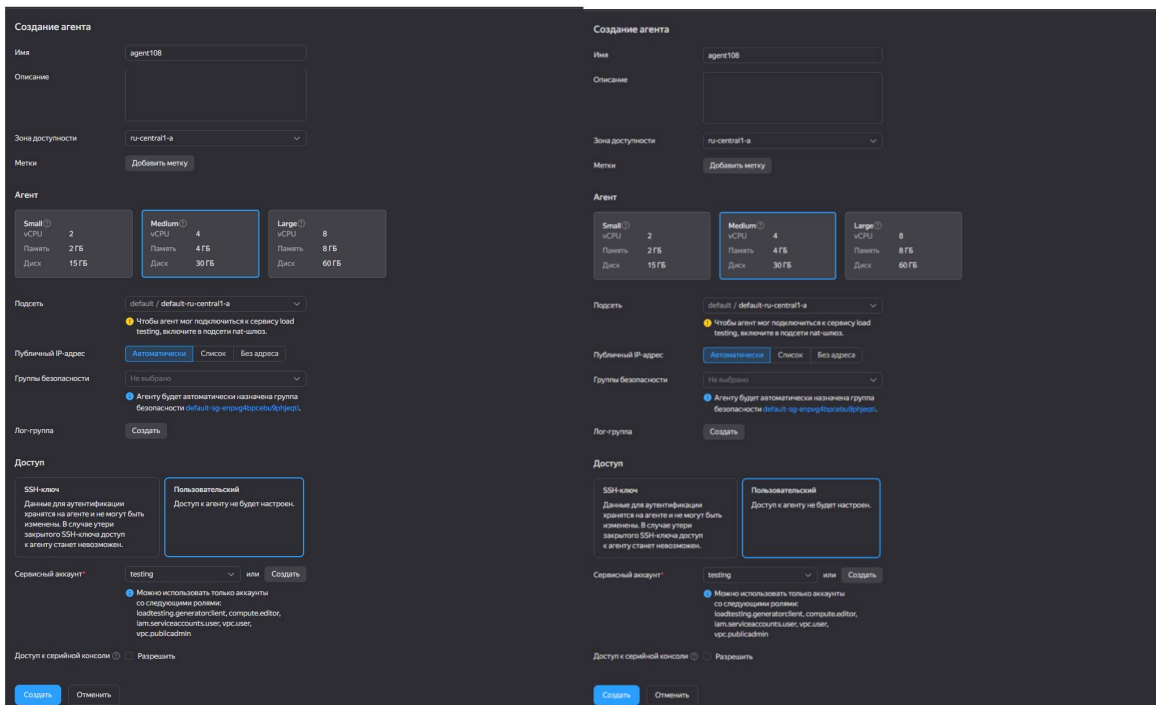


Рисунок 2. Интерфейс создания агента

Таким образом, на этапе подготовки тестового окружения формируется инфраструктурная основа для дальнейшей настройки нагрузочного теста: определяется тестируемый сервис, создается агент генерации нагрузки, настраиваются сетевые параметры и обеспечивается техническая готовность среды к проведению эксперимента.

На этапе настройки теста в Yandex Load Testing выбирается ранее созданный агент и задаются основные параметры нагрузочного тестирования.

Сервис позволяет создавать тест через интерфейс управления, где указываются цель тестирования, генератор нагрузки, параметры нагрузки и дополнительные настройки. Для первого сценария целесообразно использовать настройку через форму, так как она позволяет задать основные параметры без подготовки сложной конфигурации.

Дополнительно при настройке теста задаются тестовые данные, количество тестирующих потоков, время принудительного завершения теста и параметры мониторинга (рисунок 3).

Эти настройки необходимы для того, чтобы нагрузка формировалась контролируемо, а состояние агента можно было отслеживать в процессе выполнения теста.

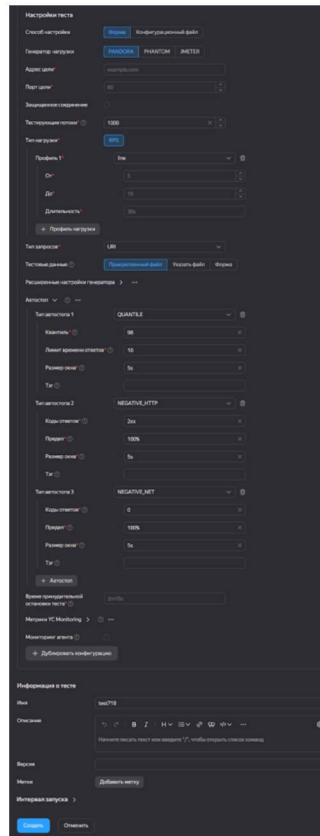


Рисунок 3. Создание и конфигурация теста

Для более полной оценки поведения информационной системы тату-салона нагрузочное тестирование целесообразно проводить не в рамках одного запуска, а в виде серии прогонов с различными параметрами нагрузки. Такой подход позволяет сопоставить поведение системы при умеренной, повышенной и постепенно возрастающей интенсивности запросов, а также подготовить данные для последующего анализа времени отклика, ошибок и устойчивости работы (таблица 3).

Таблица 3.1 – Серия прогонов нагрузочного тестирования

№ прогона	Название теста	Характер нагрузки	Назначение
1	tattoo-backend-smoke	Базовая проверка	Проверка доступности сервиса и корректности обработки запросов
2	tattoo-backend-const-low	Низкая постоянная нагрузка	Оценка работы системы в штатном режиме
3	tattoo-backend-line-growth	Линейно возрастающая нагрузка	Анализ изменения поведения сервиса при росте интенсивности запросов
4	tattoo-backend-step-load	Ступенчатая нагрузка	Выявление реакции системы на резкое увеличение нагрузки

Проведение серии прогонов также важно с точки зрения сопоставления результатов. Использование нескольких запусков с разными профилями нагрузки позволяет определить, при каких условиях начинают проявляться признаки деградации производительности, и подготовить основу для дальнейшего сравнения полученных показателей в следующей главе.

В рамках анализа результатов нагрузочного тестирования информационной системы тату-салона основное внимание уделялось поведению серверной части приложения при различных вариантах нагрузки. Для оценки использовались изменение времени отклика, характер кодов ответов и общая устойчивость работы сервиса. Такой подход позволяет рассматривать производительность системы комплексно.

При базовом сценарии сервис работает стабильно: сохраняется приемлемое время отклика и корректная обработка запросов. Это свидетельствует о том, что при невысокой интенсивности обращений серверная часть приложения справляется с нагрузкой без выраженных признаков деградации.

При постоянной низкой нагрузке система также остается устойчивой, что позволяет оценить ее работу в условиях длительной обработки однотипных запросов.

Более показательными являются сценарии с увеличением нагрузки. При линейном росте интенсивности запросов возрастает нагрузка на серверную логику и базу данных, из-за чего время отклика может постепенно увеличиваться. При ступенчатой нагрузке такие изменения проявляются еще заметнее, так как сервис должен адаптироваться к более резкому росту числа обращений.

Дополнительное значение имеет характер кодов ответов. Преобладание успешных ответов указывает на корректную работу сервиса, тогда как появление отдельных ошибок может свидетельствовать о росте нагрузки на приложение, базу данных или инфраструктуру. В совокупности с изменением времени отклика это дает общее представление о поведении серверной части проекта под нагрузкой.

Для обобщения результатов была составлена сводная таблица, отражающая поведение сервиса при разных вариантах тестирования (таблица 4).

Таблица 4. Сводные результаты нагрузочного тестирования информационной системы

Название теста	Профиль нагрузки	Время отклика	Коды ответов	Общая оценка результата
tattoo-backend-smoke	базовая проверка	стабильное, низкое	преобладают успешные ответы	сервис работает корректно, признаков перегрузки не наблюдается
tattoo-backend-const-low	постоянная низкая нагрузка	стабильное, с незначительными колебаниями	в основном успешные ответы	система сохраняет устойчивость при длительной умеренной нагрузке
tattoo-backend-line-growth	линейно возрастающая нагрузка	умеренно увеличивается по мере роста нагрузки	сохраняется преобладание успешных ответов, возможны единичные отклонения	сервис в целом устойчив, однако заметна зависимость времени отклика от интенсивности запросов
tattoo-backend-step-load	ступенчатая нагрузка	заметно возрастает на отдельных этапах	возможны отдельные ошибки при повышении нагрузки	при резком увеличении нагрузки наблюдается ухудшение производительности, что указывает на чувствительность системы к интенсивным обращениям

На основании данных, представленных в таблице, можно сделать вывод, что сервис в наибольшей степени сохраняет устойчивость при базовой и умеренной постоянной нагрузке. При более сложных сценариях, связанных с постепенным или ступенчатым увеличением интенсивности запросов, система продолжает функционировать, однако демонстрирует более выраженную зависимость времени отклика от уровня нагрузки. Это свидетельствует о том, что серверная часть приложения пригодна для работы в стандартных режимах, но при росте интенсивности обращений ее производительность начинает снижаться.

Таким образом, Yandex Load Testing можно считать подходящим средством для оценки производительности и устойчивости серверной части информационной системы тату-салона.

Заключение. Анализ сервиса Yandex Load Testing показывает, что он соответствует основным требованиям, предъявляемым к современным средствам нагрузочного тестирования. Yandex Load Testing наиболее эффективен при оценке реакции веб-сервиса на рост числа запросов, определения пределов его стабильной работы и выявления потенциальных узких мест.

Список литературы

[1] Recommendations for performance testing // Microsoft Learn. – 2025. – URL: <https://learn.microsoft.com/en-us/power-platform/well-architected/performance-efficiency/performance-test> (дата обращения: 19.03.2026).

[2] Features // Distributed Load Testing on AWS. – AWS Documentation. – URL: <https://docs.aws.amazon.com/solutions/latest/distributed-load-testing-on-aws/features.html> (дата обращения: 19.03.2026).

[3] Yandex Load Testing overview // Yandex Cloud Documentation. – 2025. – URL: <https://yandex.cloud/en/docs/load-testing/concepts/> (дата обращения: 20.03.2026).

Авторский вклад

Кошлатый Кирилл Александрович – проведение исследования методик нагрузочного тестирования, анализ теоретических основ нагрузочного тестирования, изучение особенностей облачного нагрузочного тестирования и требований к современным средствам нагрузочного тестирования, участие в подготовке выводов по применению Yandex Load Testing.

Лихтар Илья Владимирович – анализ возможностей сервиса Yandex Load Testing, рассмотрение его архитектуры, основных компонентов, преимуществ и ограничений, формулирование итоговых выводов по применению сервиса для нагрузочного тестирования веб-сервисов.

Ефремова Александра Юрьевна – руководство исследованием и постановка задачи по изучению возможностей сервиса для оценки производительности веб-сервисов.

EVALUATING WEB SERVICE PERFORMANCE USING YANDEX LOAD TESTING

K.A. Koshlatyi

*Student of the department
Information and Computer-
Aided Systems Design, BSUIR*

I.V. Likhtar

*Student of the department
Information and Computer-
Aided Systems Design, BSUIR*

A. Yu. Yafremava

*Assistant of the department
Information and Computer-Aided
Systems Design, BSUIR*

Abstract. The article examines the use of Yandex Cloud Testing as an effective tool for evaluating the performance of web services.

Keywords: testing, test cases, optimization of design solutions, cloud testing.