

УДК 004.432.2

Исследование реализации микросервисной архитектуры с использованием языка программирования Python

Иванов А.С. (ИТМО)

Научный руководитель – кандидат физико-математических наук, Жуков Н.Н. (ИТМО)

Введение. В условиях динамично развивающихся технологий повышается интерес к микросервисам как к гибкому решению для разработки программного обеспечения. При реализации микросервисной архитектуры активно используется Python, являющийся одним из наиболее востребованных специалистами языков, в том числе за счет простоты и наличия разнообразных библиотек и инструментов [1]. Тем не менее, существующие исследования в сфере микросервисной архитектуры зачастую акцентируют внимание в большей степени на различных архитектурных подходах, инструментах CI/CD, а также методах контейнеризации. При этом эффективность языков и фреймворков, разработанных на них, остается в стороне, что затрудняет выбор оптимальных технологий для реализации микросервисов [2]. Создание микросервисного приложения предполагает выбор ключевых компонентов и их конфигураций, разработку стратегии взаимодействия между сервисами и проработку инфраструктуры для управления и мониторинга системы. Такой подход позволяет упростить поддержку приложения, ускорить внедрение изменений и обеспечить независимое масштабирование наиболее нагруженных компонентов. Кроме того, микросервисы удобны при использовании контейнеризации и современных инструментов оркестрации, например, Docker и Kubernetes, что обеспечивает портативность и гибкость развертывания [3].

Основная часть. В ходе исследования, на примере разработанного приложения-образца, проведены измерения времени отклика и устойчивости используемых в нём микросервисов, что позволило получить объективные данные о производительности системы. Результаты тестирования продемонстрировали высокую эффективность и высокий процент успешных запросов при больших нагрузках, что в данном случае свидетельствует о надежности микросервисного подхода. На основе собранных экспериментальных данных была разработана модель оценки паттернов, используемых при проектировании и реализации приложения, которая послужила инструментом для анализа работы микросервисов. Эта модель позволила обобщить результаты экспериментов и выявить как сильные стороны системы, так и области, требующие оптимизации. Основными аспектами для мониторинга и анализа стали скорость обработки запросов, время отклика сервисов и уровень ресурсоемкости, которые предоставляют возможность глубже понять динамику работы микросервисной архитектуры. Проведенный анализ раскрыл новые возможности для повышения эффективности приложения за счет оптимизации работы выбранных микросервисов.

Выводы. Исследование предоставляет практический инструментарий для оценки и выбора технологий при проектировании микросервисных архитектур. Полученные данные позволяют восполнить недостаток работ, фокусирующихся исключительно на инфраструктурных аспектах, дают разработчикам возможность принимать обоснованные решения на основе количественных метрик эффективности языков и фреймворков. Это открывает путь к созданию более сбалансированных и производительных приложений, соответствующих требованиям современных высоконагруженных систем.

Список использованных источников:

1. Rodionov, K. CONTAINERIZING PYTHON APPLICATIONS USING DOCKER TO BUILD A MICROSERVICE ARCHITECTURE / K. Rodionov. — Текст : электронный //

Научные журналы Universum : [сайт]. — URL:
<https://7universum.com/ru/tech/archive/item/17827> (дата обращения: 15.02.2025).

2. Ногоева, Г. Д. Создание веб приложений посредством библиотеки языка программирования Python / Г. Д. Ногоева, К. Д. Абыкеев // Вестник Кыргызстана. – 2021. – № 1-2. – С. 179-182. – DOI 10.33514/ВК-1694-7711-2021-1(2)-179-182. – EDN LTIKNB.

3. Крис Р. Микросервисы. Паттерны разработки и рефакторинга. – "Издательский дом" "Питер", 2023. №1 С. 119-126