УДК 004.421, 004.75

СТАТИЧЕСКИЙ ПОДХОД К ВИЗУАЛИЗАЦИИ ВЗАИМОДЕЙСТВИЯ МИКРОСЕРВИСОВ И ИНФРАСТРУКТУРНЫХ КОМПОНЕНТОВ

Филиппов А.А. (ИТМО)

Научный руководитель - кандидат технических наук, инженер факультета инфокоммуникационных технологий Харитонов А.Ю. (ИТМО)

Введение. В последнее десятилетие микросервисная архитектура стала одной из доминирующих парадигм разработки программного обеспечения [2][3]. Хотя микросервисы решили многие проблемы, характерные для монолитных систем, их децентрализованная природа существенно затрудняет создание и поддержание целостного представления о системе. Без комплексного инструмента визуализации архитекторам, разработчикам и администраторам сложно отслеживать взаимодействия между сервисами, выявлять узкие места и оптимизировать производительность системы. Даже если отдельные компоненты функционируют корректно, проблемы часто возникают именно на уровне их взаимодействия. С течением времени в системе могут накапливаться неэффективности, а фактическая архитектура может существенно отклоняться от изначально спроектированной. Однако практически вся необходимая для визуализации информация обычно уже содержится в самом проекте или инфраструктурных компонентах, будь то конфигурационные файлы, файлы миграций или исходный код. Но эти сырые данные сложны для восприятия и анализа без соответствующего графического представления. Система визуализации взаимодействия микросервисов и инфраструктурных компонентов, как раз и занимается отображением структуры системы, а также поддержанием ее в актуальном состоянии, помогая быстро идентифицировать проблемные области и принимать обоснованные архитектурные решения. В данной работе же представлен комплексный подход к созданию системы визуализации основанной на статическом подходе, анализирующим исходный код проектов для формирования схемы взаимодействия.

Основная часть. В ходе анализа существующих решений выявлено, что порядка 90% современных утилит используют динамический подход со сбором данных при помощи анализа трейсов, однако они имеют свои минусы такие как большая дополнительная нагрузка на инфраструктуру и необходимость в продолжительной работе для предоставления полной информации о системе [1]. Статический подход так или иначе используют только 2 решения, а именно MicroArt, но и они оказались недостаточными для использования в качестве полноценного визуализатора. Из-за чего было принято решение разработки собственного решения, реализующего все достоинства статического анализа и минимизирующего его недостатки. Представляемая модель системы состоит из следующих компонентов: источник данных – предоставляет первоначальную информацию об инфраструктуре, сборщик данных – собирает и обрабатывает данные, полученные из источников и отправляет их в хранилище, хранилище данных – хранит преобразованные данных источников в виде графа визуализации, обновитель данных – обновляет данные в хранилище после их изменения в источниках данных и визуализатор данных – предоставляет визуальный интерфейс представления данных из хранилища. Данная модель была реализована с помощью языка Golang, где в качетств источников использовались: Yaml файлы, Gitlab API, файлы миграций и PostgreSQL API. После чего протестирована на инфраструктуре состоящей из 89 сервисов.

Выводы. Был проведен комплексный анализ существующих подходов к визуализации микросервисной архитектуры, который выявил преобладание динамических методов анализа над статическими. Для восполнения данного пробела было разработано собственное решение, основанное на статическом анализе и реализованное на языке Golang, продемонстрировавшее высокую эффективность на выборке из 89 микросервисов. Система осуществила построение полной схемы взаимодействий за 9 минут при минимальном потреблении ресурсов (не более 10.1 MB оперативной памяти и 0.7 CPU). В сравнении с

динамическими подходами, предложенное решение показало значительную экономию вычислительных ресурсов за счет использования единого экземпляра приложения и базы данных вместо множества распределенных агентов сбора данных. Кроме того практическое внедрение системы позволило выявить ряд существенных проблем в архитектуре исследуемых проектов, включая проблемы с миграциями, высокую связность компонентов и неоптимальное распределение нагрузки на базы данных, а сгенерированная схема архитектуры успешно послужила документацией при миграции между брокерами сообщений и помогла в подготовке нагрузочного тестирования путем выявления всех используемых внешних API. Результаты исследования демонстрируют эффективность статического подхода к визуализации микросервисной архитектуры и открывают перспективы для дальнейшего развития инструментов архитектурного анализа распределенных систем. Предложенное решение может быть адаптировано под различные технологические стеки и масштабы проектов, предоставляя разработчикам и архитекторам эффективный инструмент для управления сложностью современных микросервисных систем.

Список использованных источников:

1. M. E. Gortney et al. Visualizing Microservice Architecture in the Dynamic Perspective: A Systematic Mapping Study // IEEE Access, vol. 10, pp. 119999-120012, 2022
2. Cloud Microservice Market Outlook from 2024 to 2034: [электронный ресурс]. – URL: https://www.futuremarketinsights.com/reports/cloud-microservice-market (дата обращения 14.12.2024)
3. StackOverflow 2024 Developer Survey: [электронный ресурс]. – URL: https://survey.stackoverflow.co/2024/technology (дата обращения 14.12.2024)

Автор _____ Филиппов А.А.

Научный руководитель _____ Харитонов А.Ю.