



УДК 004.273

Н. Т. СУХАНОВА, канд. пед. наук, доц. кафедры информационных систем и технологий; **А. А. ПАВЛОВ**, магистрант кафедры информационных систем и технологий

РЕШЕНИЕ ПРОБЛЕМЫ ПСЕВДОСИНХРОННОГО ВЗАИМОДЕЙСТВИЯ В СИСТЕМАХ ЦИФРОВОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА ИЗДЕЛИЙ

ФГБОУ ВО «Нижегородский государственный архитектурно-строительный университет»
Россия, 603952, г. Н. Новгород, ул. Ильинская, д. 65.

Тел.: (831) 430-54-92; факс: (831) 430-19-36; эл. почта: ntsuhanova@gmail.com

Ключевые слова: псевдосинхронное взаимодействие, синхронное взаимодействие, асинхронное взаимодействие, архитектурная уязвимость, микро-сервисная архитектура, архитектурные паттерны, событийно-ориентированная архитектура.

В статье рассматриваются проблемы псевдосинхронного взаимодействия в архитектуре информационных систем. Вводится понятие псевдосинхронного взаимодействия. Приводится анализ источников по теме исследования. Уделяется внимание причинам возникновения псевдосинхронного взаимодействия, теоретическим и практическим способам их предотвращения. Рассматриваются практические примеры устранения сложностей, связанных с вопросами псевдосинхронного взаимодействия. Приводятся способы применения комплексных подходов решения проблемы псевдосинхронного взаимодействия в инженерных информационных системах.

Введение. В условиях стремительного развития информационных и коммуникационных технологий архитектура информационных систем становится одним из ключевых аспектов успешного функционирования организации. При этом в современных реалиях структура приложений подразумевает частое межсервисное или вовсе более глобальное межсистемное взаимодействие. Благодаря этому, а также вследствие разнообразия в методологиях, способах и подходах к работе каждого отдельного разработчика или системного архитектора, нехватки времени или иных ресурсов порой возникает интересный недочет в архитектуре информационных систем, который будем называть псевдосинхронным взаимодействием.

В данной статье под псевдосинхронным взаимодействием подразумевается особая ситуация, когда различные компоненты системы обмениваются данными или управляют друг другом таким образом, что на первый взгляд создается ощущение синхронности, хотя на самом деле взаимодействие происходит асинхронно. По своей сути, можно рассматривать этот аспект как граничное пересечение данных типов общения. Это может быть связано с временными задержками в передаче данных, разными обработчиками событий или спецификой работы сетевых протоколов. В итоге пользователи могут не осознавать, что фактически взаимодействие происходит не в реальном времени, что может приводить к искажениям и задержкам в обработке информации.



Проблема псевдосинхронного взаимодействия затрагивает как инженерные информационные системы, в том числе ТИМ, СОД, СУИД и пр., так и информационные системы бизнес-процессов.

Особую значимость решение проблемы псевдосинхронного взаимодействия имеет в рамках разработки инженерных информационных систем, включающих построение средств цифрового документирования, электронных архивов технической документации, а также в процессе создания связи с изготовителем и потребителем изделий.

Цель данной статьи – проанализировать проблему псевдосинхронного взаимодействия в архитектуре инженерных информационных систем, выявить ее причины, последствия и предложить рекомендации по улучшению архитектурных решений. Для этого необходимо решить следующие задачи:

- определить ключевые характеристики и параметры, которые позволяют идентифицировать псевдосинхронное взаимодействие в системах;
- исследовать основные причины возникновения псевдосинхронности, включая особенности проектирования и реализации систем;
- рассмотреть влияние псевдосинхронного взаимодействия на пользовательский опыт и функционирование приложений;
- разработать рекомендации для системных архитекторов и разработчиков по минимизации рисков, связанных с псевдосинхронным взаимодействием;
- детально рассмотреть несколько практических вариантов решения конкретного случая проблемы в архитектуре двух взаимодействующих между собой информационных систем.

Проблема псевдосинхронного взаимодействия может оказать существенное влияние на производительность, надежность и управляемость информационных систем. Особенно тяжело приходится другим сервисам, которые так или иначе вынуждены обращаться к псевдосинхронным частям таких систем. Недостаточное внимание к деталям их работы и недочеты некоторых особенностей в процессе разработки взаимодействия с псевдосинхронными участками могут легко привести к краху всех взаимосвязанных систем, что может также повлечь за собой и более серьезные проблемы уже вне технологического поля.

Проблема существующих исследований. Рассмотрение архитектурных уязвимостей и способов их решения в информационных системах поднимается довольно часто. Особенно это характерно для набирающей популярность микро-сервисной архитектуры, в рамках которой по большей части и возникает рассматриваемая проблема из-за необходимости взаимодействия между собой разных сервисов. Так М. Н. Карпович в своей работе рассматривает как преимущества, так и недостатки микро-сервисной архитектуры, при этом затрагивая шаблоны проектирования, методики разработки и даже интересующие в рамках данного исследования режимы взаимодействия сервисов [1]. В работе упоминается и даже анализируется асинхронное взаимодействие с помощью брокеров сообщений, однако этого недостаточно для решения проблемы.

О. А. Тимакина, А. В. Зиборева рассматривают синхронно-асинхронные взаимодействия, однако в работах уделяется внимание только одному типу взаимодействия [2, 3]. Авторы затрагивают основные принципы синхронного общения, однако в них отсутствует глубокий анализ сложных сценариев, связанных с непосредственным использованием данного типа взаимодействия, не



говоря уже о рассмотрении альтернативы в виде асинхронной связи. Особого внимания заслуживает статья К. С. Глумова, в которой он подробно рассматривает асинхронный вид взаимодействия на примере реактивного программирования, которое по своей сути является более частной формой данного типа общения [4]. Работа вполне самостоятельна и хорошо раскрывает тему, но она также затрагивает лишь один тип взаимодействия, рассматривая его практическое применение в общем виде, без каких-либо граничных моментов и исключений. Аналогично, недосказанность в виде рассмотрения лишь асинхронного типа взаимодействия присутствует и в работах Е. А. Тришина, Н. А. Ивановой и А. М. Сотченкова [5, 6].

Как показал анализ источников, большая часть из них рассматривает преимущества и недостатки синхронного и асинхронного взаимодействия между информационными системами, приводятся общие примеры их реализации [7, 8]. То есть, исследуются лишь основные принципы двух типов взаимодействия без какого-либо глубокого анализа сложных сценариев. В этих источниках не уделяется внимание отклонениям от стандартных формулировок данных режимов взаимодействия, которым и является псевдосинхронный тип общения сервисов. Неполнота информации большинства рассмотренных работ указывает на актуальность темы исследования.

Разновидности и причины появления проблемы. Существует множество разновидностей рассматриваемой проблемы и, как следствие, причин их появления. Одной из самых заметных является разница в скорости обработки данных. Компоненты системы могут иметь разные временные характеристики: один модуль может обрабатывать запросы мгновенно, в то время как другой может требовать больше времени из-за сложности операций или задержек в доступе к ресурсам. Эта разница заставляет один компонент ожидать ответа от другого, что и создает временной разрыв и недопонимание среди пользователей.

Кроме того, современные информационные системы часто функционируют в распределенной среде, что подразумевает наличие множества компонентов, расположенных на разных серверах или в различных географических точках. При взаимодействии между такими компонентами сетевые задержки становятся неизбежными, поскольку время, необходимое для передачи данных, может варьироваться. Даже в высокоскоростных сетях наличие задержек может привести к проблемам синхронизации и взаимодействия между элементами системы.

Асинхронные модели взаимодействия, которые становятся все более популярными в архитектуре микро-сервисов и событийно-ориентированного программирования, также могут способствовать развитию псевдосинхронного поведения [1]. Такие модели часто проектируются в целях повышения гибкости и масштабируемости, и хотя они могут быть весьма эффективными, они порой создают ложное ощущение синхронности. Когда один сервис отправляет запрос другому и продолжает свою работу без ожидания ответа, может складываться впечатление, что система функционирует без задержек, однако реальность такова, что фактические задержки остаются незамеченными.

Использование кэширования данных, направленного на ускорение доступа и оптимизации нагрузки на базу данных, также может привести к возникновению псевдосинхронного взаимодействия. Кэш может содержать устаревшие данные, если информация в основной базе данных была обновлена, но изменения не



синхронизировались с кэшем. Это ведет к тому, что компоненты, полагающиеся на кэш для получения информации, могут работать с некорректными данными, создавая впечатление синхронности, которая на деле отсутствует.

Другим важным аспектом является проблема согласованности данных. Когда несколько компонентов одновременно работают с одной и той же информацией, это может привести к конфликтам и несоответствиям. В распределенных системах обновления данных могут происходить в одном месте, но не синхронизироваться с другими частями системы, что создает временные несоответствия и вызывает трудности во взаимодействии между пользователями и компонентами. Непредсказуемые сбои в работе системы, такие как проблемы с сетью или отказ аппаратного обеспечения, также могут способствовать появлению псевдосинхронного поведения. Когда часть системы становится недоступной, это создает временные задержки, которые не всегда очевидны для пользователей и других компонент, продолжающих попытки взаимодействия с недоступным модулем.

Стратегии по предотвращению проблемы. Для минимизации негативных последствий псевдосинхронного взаимодействия в информационных системах можно применить ряд стратегий, способствующих повышению их эффективности и надежности. Важным шагом является улучшение мониторинга системы: внедрение систем логирования и отслеживания времени отклика поможет выявить узкие места и определить, где возникают задержки. Оптимизация архитектуры также играет ключевую роль в решении данной проблемы. Использование архитектурных паттернов, таких как *CQRS* (разделение ответственности команд и запросов) или *Event Sourcing* (сохранение состояния системы в виде событий), позволяет более эффективно управлять состоянием и согласованностью данных, что, в свою очередь, снижает вероятность возникновения псевдосинхронных ситуаций.

Кроме того, стоит рассмотреть возможность применения асинхронных методов программирования. Замена синхронных вызовов на асинхронные может разгрузить систему и значительно повысить ее производительность, позволяя компонентам работать более эффективно и независимо друг от друга. Важно также не забывать об обучении пользователей: понимание принципов работы системы и наличие эффективных механизмов обратной связи помогут пользователям легче ориентироваться в ее функционале и состояниях процессов, что снизит их тревожность в ситуациях, когда происходит задержка.

Практические примеры решения проблемы. Рассмотрим небольшой практический пример, где проблема псевдосинхронного взаимодействия возникла из-за неправильного выбора типа общения между системами при планировании архитектуры. Перед нами предстанет довольно простой сервис формирования отчетов, к которому время от времени обращается сторонний для их получения, общая схема данного сервиса представлена на рис. 1. В текущей реализации процесс занимает минимум времени и работает по принципу синхронного взаимодействия, по своей сути блокируя процесс до возвращения отчета как ответа на запрос сторонней системы [2].

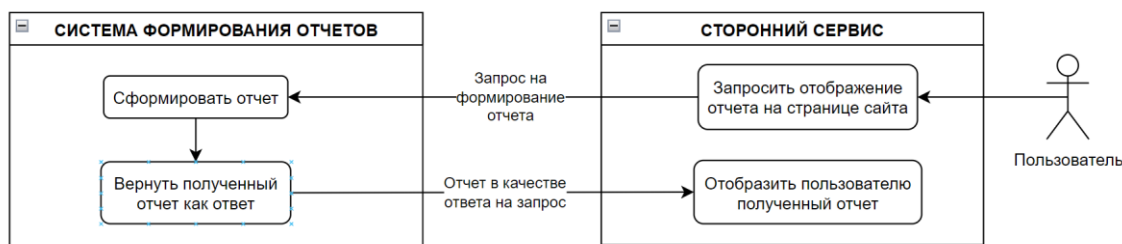


Рис. 1. Процессы, протекающие в инженерных информационных системах формирования отчета

Однако, со временем состав данного отчета и количество данных в нем только увеличивается, вследствие чего растет и время ожидания его формирования. Появляется потребность в изменении логики работы данного процесса, теперь изначальный запрос лишь инициирует старт действия и в ответ посылает информацию об успешности его запуска. Для получения результата или же просто проверки статуса теперь используется отдельный запрос. Общая схема новой реализации в системе формирования отчетов для лучшего понимания представлена с помощью спецификации *ВРМН* на рис. 2. При этом стоит отметить, что проверки вводных данных, обработка ошибок и прочие мелкие моменты опущены для упрощения вида диаграммы.

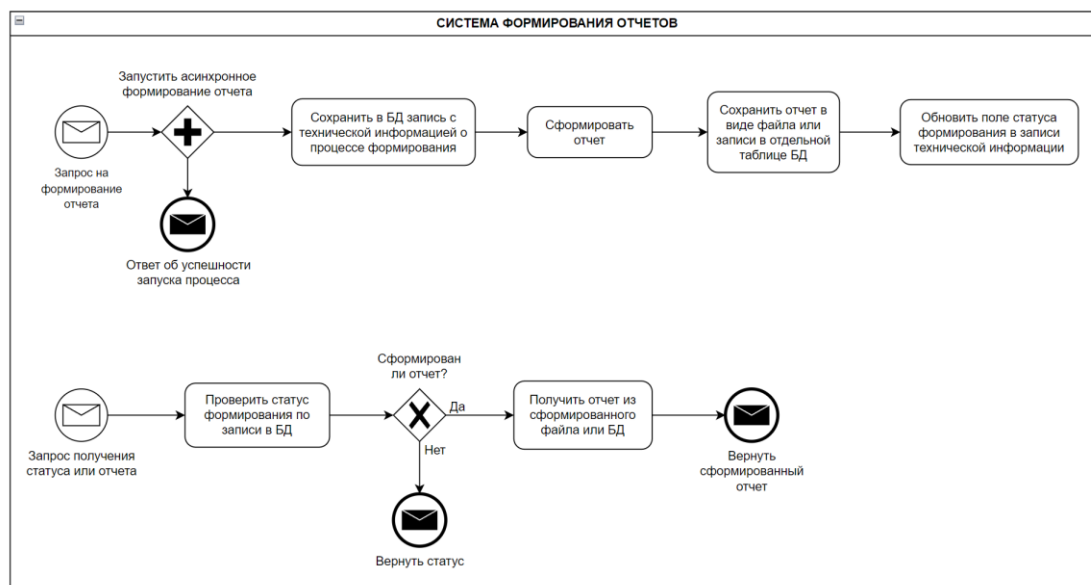


Рис. 2. Псевдосинхронный процесс формирования отчета

Таким образом, получается реализовать псевдосинхронное взаимодействие, когда мы, с одной стороны, общаемся синхронным способом, ожидая какую-либо информацию, хотя, с другой стороны, вынуждены постоянно проверять ее появление дополнительными, скорее всего неоднократными запросами. Текущее архитектурное решение повлечет за собой внушительные изменения в смежных системах, которым необходимо получение упомянутых отчетов. Чаще всего подобная ситуация возникает из-за отсутствия преждевременного планирования и

расчета масштабирования системы и ее логики в дальнейшем, а также отсутствия времени и ресурсов на изменение уже существующей архитектуры на более подходящую.

Пути решения рассматриваемой проблемы. Есть несколько путей решения данной разновидности рассматриваемой проблемы, в зависимости от количества времени, ресурсов и возможности изменения системы формирования отчетов. Поскольку порой мы можем выступать исключительно со стороны «стороннего» сервиса без возможности влияния на остальные.

Первым и наиболее правильным вариантом в таком случае будет перевод взаимодействия с сервисом формирования отчетов на полностью асинхронное. Сделать это можно посредством использования брокеров сообщений, архитектуры и способов применения, которые рассматривает в своей работе Тришин Е. А., сравнивая несколько самых популярных подобных инструментов [5]. Они представляют собой промежуточный сервис хранения и управления сообщениями, которые уже будут отправляться и читаться определенными системами.

Архитектура самих приложений будет при этом сильно похожа на последнюю рассмотренную, поскольку нам все также приходится сначала отправить запрос в сервис отчетов, специальный слушатель на нашей стороне должен дождаться ответного сообщения, и уже по отдельному запросу внешнее представление будет получать данные в отчете. Единственное исключение заключается в том, что теперь система формирования отчетов лишь выполняет сам основной процесс, а все остальные взаимодействия происходят уже на стороне прочих сервисов. Общая схема их работы при таком варианте представлена на рис. 3, 4.

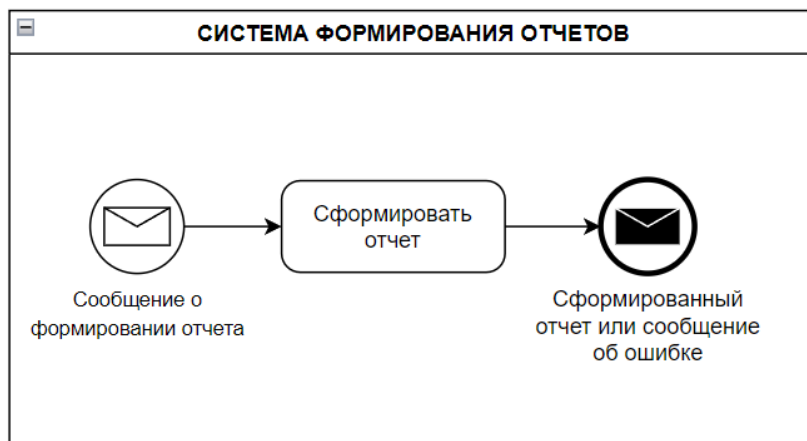


Рис. 3. Работа системы формирования отчетов при задействовании брокера сообщений

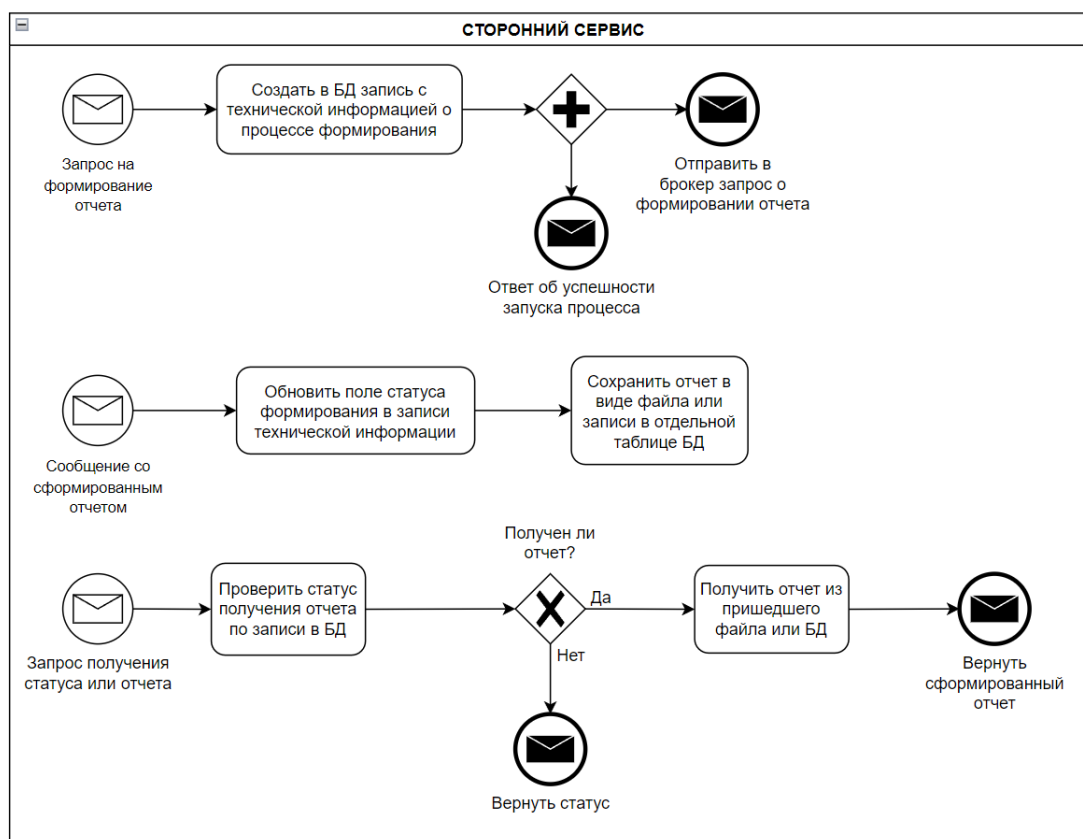


Рис. 4. Работа сторонних систем при взаимодействии с сервисом формирования отчетов с помощью брокера сообщений

Однако, довольно часто происходит так, что мы не можем повлиять на исходную систему и тем более ее архитектуру. В таком случае, сервис формирования отчетов у нас будет представлен все той же схемой, представленной на рис. 2. Сторонние же системы будут вынуждены адаптироваться к этим изменениям путем доработки уже своей архитектуры, в частности, добавлением множественной асинхронной отправки запросов для получения текущего статуса процесса. На рис. 5 подробно представлена реализация обработки подобной логики со стороны других сервисов.

По своей сути мы просто грамотно обрабатываем недостаток архитектуры системы формирования отчетов, выполняем множественное обращение к ней с заданным временным промежутком. При этом ничего не мешает дополнить реализацию ограничением по количеству подобных повторяющихся запросов, после достижения которых возможно отображение для пользователя информационного сообщения о неудаче.

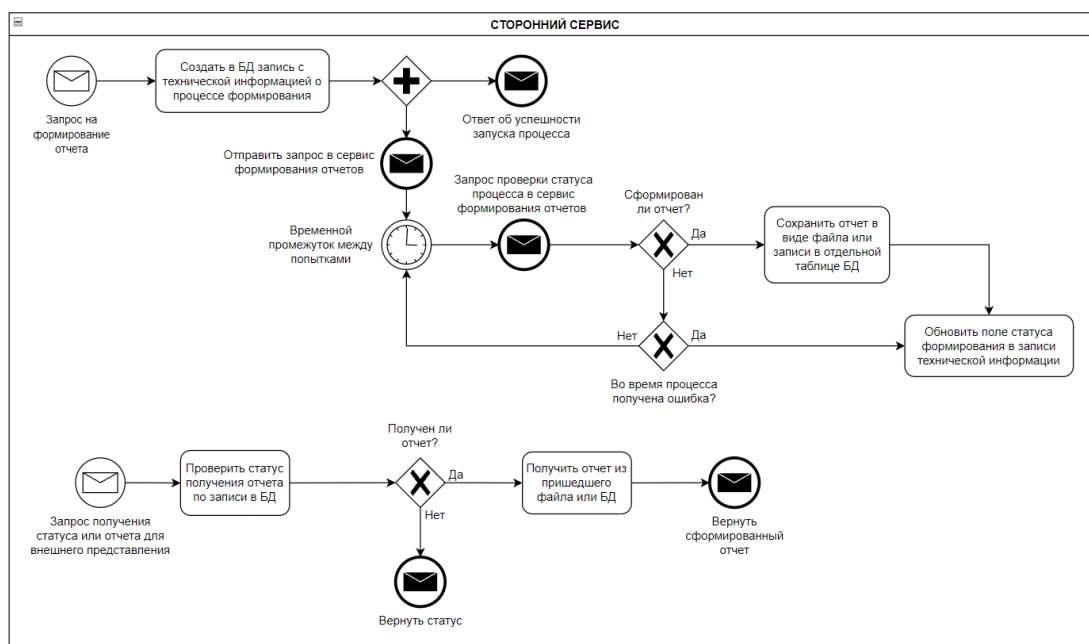


Рис. 5. Обработка псевдосинхронного взаимодействия

Заключение. Проблема псевдосинхронного взаимодействия в архитектуре инженерных информационных систем, которые являются основными инструментами цифровой поддержки жизненного цикла изделий, представляет собой значительный вызов современным подходам к проектированию и реализации IT-решений. Псевдосинхронность может привести к различным неисправностям, замедлению обработки данных и ухудшению качества обслуживания пользователей. Неэффективная координация между компонентами системы может снизить общую производительность и увеличить время отклика, что критично в условиях растущих требований к скорости и надежности информационных систем.

Для решения данной проблемы необходимо применять комплексные подходы, включая использование современных архитектурных стилей, таких как микро-сервисы и событийно-ориентированную архитектуру, а также внедрение стандартов и протоколов, способствующих лучшей интеграции и взаимодействию между компонентами. Особое внимание следует уделить вопросам управления состояниями, обработки событий и мониторинга систем в реальном времени.

Таким образом, дальнейшие исследования в области псевдосинхронного взаимодействия и разработки практических методов его преодоления станут важным шагом в направлении создания более эффективных и устойчивых инженерных информационных систем, способных справляться с вызовами цифровой эпохи. Это позволит организациям реализовывать свои бизнес-цели и предлагать качественные услуги, соответствующие ожиданиям пользователей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Карпович, М. Н. Особенности проектирования микросервисно-событийных архитектур для высоконагруженных распределенных систем обработки информации / М.



Н. Карпович. – Текст : непосредственный // Труды БГТУ. Серия 3. Физико-математические науки и информатика. – 2023. – № 1 (266). – С. 89–95.

2. Тимакин, О. А. Организация синхронного взаимодействия для интеграции информационных систем / О. А. Тимакин. – Текст : непосредственный // Евразийский научный журнал. – 2017. – № 1. – С. 19–23.

3. Зиборов, А. В. Антипаттерны построения микросервисных приложений в высоконагруженных проектах. – Текст : непосредственный // Universum : технические науки. – 2023. – № 11–1 (116). – С. 29–34.

4. Глузов, К. С. Преимущества реактивного подхода в высоконагруженных микросервисных системах / К. С. Глузов. – Текст : непосредственный // Вестник науки. – 2024. – № 10 (79). – С. 410–426.

5. Тришин, Е. А. Исследование брокеров сообщений в приложениях с микросервисной архитектурой / Е. А. Тришин. – Текст : электронный // Вестник науки : международный научный журнал. – 2024. – Том 3, № 6 (75). – С. 1469–1474. – URL: <https://www.вестник-науки.рф/article/16086>.

6. Иванова, Н. А. Организация взаимодействия между компонентами микросервисной архитектуры системы контроля и управления доступом / Н. А. Иванова, А. М. Сотченков. – Текст : непосредственный // Ученые записки Брянского государственного университета. – 2024. – № 2 (34). – С. 15–19.

7. Синхронное vs Асинхронное : выбираем подход к взаимодействию микросервисов. – URL: <https://vc.ru/u/1788045-aleksei-solonkov/709274-sinhronnoe-vs-asinhronnoe-vybiraem-podhod-k-vzaimodeistviyu-mikroservisov> (дата обращения: 07.12.24). – Текст : электронный.

8. Зачем нужны очереди сообщений в микросервисной архитектуре : разбираем преимущества и недостатки. – URL: <https://cloud.vk.com/blog/zachem-nuzhny-ocheredi-soobshcheniy-v-mikroservisnoy-arkhitekture/> (дата обращения: 07.12.24). – Текст : электронный.

SUKHANOVA Nadezhda Timofeevna, candidate of pedagogical sciences, associate professor, associate professor of the chair of information systems and technologies; PAVLOV Aleksey Andreevich, master degree student of the chair of information systems and technologies

SOLVING THE PROBLEM OF PSEUDOSYNCHRONOUS INTERACTION IN DIGITAL PRODUCT LIFECYCLE SUPPORT SYSTEMS

Nizhny Novgorod State University of Architecture and Civil Engineering

65, Пjinskaya St., Nizhny Novgorod, 603952, Russia.

Tel.: +7 (831) 430-54-92; fax: +7 (831) 430-19-36; e-mail: ntsuhanova@gmail.com

Key words: pseudosynchronous interaction, synchronous interaction, asynchronous interaction, architectural vulnerability, microservice architecture, architectural patterns, event-driven architecture.

The article considers the problems of pseudosynchronous interaction in the architecture of information systems. The concept of pseudosynchronous interaction is introduced. An analysis of sources on the topic of the study is given. Attention is paid to the causes of pseudosynchronous interaction, theoretical and practical ways to prevent them. Practical examples of eliminating the difficulties associated with pseudosynchronous interaction issues are considered. Methods for applying complex approaches to solving the problem of pseudosynchronous interaction in engineering information systems are presented.



REFERENCES

1. Karpovich M. N. Osobennosti proektirovaniya mikroservisno-sobytiynykh arkhitektur dlya vysokonagruzhennykh raspredelennykh sistem obrabotki informatsiy [Features of designing microservice-event architectures for high-loaded distributed information processing systems]. Trudy BGTU. Seriya 3. Fiziko-matematicheskie nauki i informatika [Proceedings of BSTU. Series 3. Physical and Mathematical Sciences and Informatics], 2023, № 1 (266), P. 89–95.
2. Timakin O. A. Organizatsiya sinkhronnogo vzaimodeystviya dlya integratsii informatsionnykh sistem [Organization of synchronous interaction for the integration of information systems]. Evraziyskiy nauchnyy zhurnal [Eurasian Scientific Journal], 2017, № 1, P. 19–23.
3. Ziborev A. V. Antipatterny postroeniya mikroservisnykh prilozheniy v vysokonagruzhennykh proektakh [Antipatterns of building microservice applications in high-loaded projects]. Universum : tekhnicheskie nauki [Universum : Technical Sciences], 2023, № 11–1 (116), P. 29–34.
4. Glumov K. S. Preimushchestva reaktivnogo podkhoda v vysokonagruzhennykh mikroservisnykh sistemakh [Advantages of the reactive approach in high-loaded microservice systems]. Vestnik nauki [Bulletin of Science], 2024, № 10 (79), P. 410–426.
5. Trishin E. A. Issledovanie brokerov soobshcheniy v prilozheniyakh s mikro-servisnoy arkhitekturoy [Research of message brokers in applications with micro-service architecture]. Vestnik nauki : mezhdunarodnyy nauchnyy zhurnal [Bulletin of Science: International Scientific Journal], 2024, Vol. 3, № 6 (75), P. 1469–1474. URL: <https://www.вестник-науки.рф/article/16086>.
6. Ivanova N. A., Sotchenkov A. M. Organizatsiya vzaimodeystviya mezhdru komponentami mikroservisnoy arkhitektury sistemy kontrolya i upravleniya dostupom [Organization of interaction between components of the microservice architecture of the access control and management system]. Uchenye zapiski Bryanskogo gosudarstvennogo universiteta [Scientific Notes of Bryansk State University], 2024, № 2 (34), P. 15–19.
7. Sinkhronnoe vs Asinkhronnoe : vybiraem podkhod k vzaimodeystviyu mikroservisov [Synchronous vs Asynchronous : choosing an approach to microservice interaction]. URL: <https://vc.ru/u/1788045-aleksei-solonkov/709274-sinhronnoe-vs-asinhronnoe-vybiraem-podhod-k-vzaimodeistviyu-mikroservisov> (accessed: 07.12.24).
8. Zachem nuzhny ocheredi soobshcheniy v mikroservisnoy arkhitekture : razbiraem preimushchestva i nedostatki [Why message queues are needed in a microservice architecture: analyzing advantages and disadvantages]. URL: <https://cloud.vk.com/blog/zachem-nuzhny-ocheredi-soobshcheniy-v-mikroservisnoy-arkhitekture/> (accessed: 07.12.24).

© Н. Т. Суханова, А. А. Павлов, 2025

Получено: 21.02.2025 г.