
НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

№ 3 2026

*Журнал «Научное обозрение. Технические науки»
зарегистрирован Федеральной службой
по надзору в сфере связи, информационных
технологий и массовых коммуникаций.
Свидетельство ПИ № ФС77-57440
выдано 27.03.2014.
ISSN 2500-0799*

**Двухлетний импакт-фактор РИНЦ – 0,494
Пятилетний импакт-фактор РИНЦ – 0,260**

*Учредитель, издатель и редакция:
ООО НИЦ «Академия Естествознания»,*

*Почтовый адрес: 101000, г. Москва, а/я 47
Адрес учредителя, издателя: 410056, г. Саратов,
ул. им. Чапаева В.И., д. 56
Адрес редакции: 410035, г. Саратов,
ул. Мамонтовой, д. 5*

**Founder, publisher and edition:
LLC SPC Academy of Natural History,**

**Post address: 101000, Moscow, p.o. box 47
Founder's, publisher's address: 410056, Saratov,
56 Chapaev V.I. str.
Editorial address: 410035, Saratov,
5 Mamontovoi str.**

*Подписано в печать 30.09.2026
Дата выхода номера 30.10.2026
Формат 60×90 1/8*

*Типография
ООО НИЦ «Академия Естествознания»,
410035, г. Саратов, ул. Мамонтовой, д. 5*

**Signed in print 30.09.2026
Release date 30.10.2026
Format 60×90 8.1**

**Typography
LLC SPC «Academy Of Natural History»
410035, Saratov, 5 Mamontovoi str.**

*Технический редактор Доронкина Е.Н.
Корректор Галенкина Е.С., Дудкина Н.А.*

*Распространяется по свободной цене
Тираж 100 экз. Заказ НО 2026/3*

© ООО НИЦ «Академия Естествознания»

Журнал «НАУЧНОЕ ОБОЗРЕНИЕ» выходил с 1894 по 1903 год в издательстве П.П. Сойкина. Главным редактором журнала был Михаил Михайлович Филиппов. В журнале публиковались работы Ленина, Плеханова, Циолковского, Менделеева, Бехтерева, Лесгафта и др.

Journal «Scientific Review» published from 1894 to 1903. P.P. Soykin was the publisher. Mikhail Filippov was the Editor in Chief. The journal published works of Lenin, Plekhanov, Tsiolkovsky, Mendeleev, Bekhterev, Lesgaft etc.



М.М. Филиппов (M.M. Philippov)

С 2014 года издание журнала возобновлено
Академией Естествознания

**From 2014 edition of the journal resumed
by Academy of Natural History**

Главный редактор: М.Ю. Ледванов
Editor in Chief: M.Yu. Ledvanov

Редакционная коллегия (Editorial Board)

Н.Ю. Стукова (N.Yu. Stukova)

М.Н. Бизенкова (M.N. Bizenkova)

Н.Е. Старчикова (N.E. Starchikova)

Т.В. Шнуровозова (T.V. Shnurovozova)

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

<https://science-engineering.ru>

2026 г.



***В журнале представлены научные обзоры,
статьи проблемного
и научно-практического характера***

***The issue contains scientific reviews,
problem and practical scientific articles***

СОДЕРЖАНИЕ

Технические науки

СТАТЬЯ

АНТИФРОД-СИСТЕМА: РАЗРАБОТКА ПРОГРАММНОГО
КОМПЛЕКСА ДЛЯ ПРЕДОТВРАЩЕНИЯ МОШЕННИЧЕСКИХ
ТРАНЗАКЦИЙ

Дубровский А. А., Малёшина Л. М. 5

НАУЧНЫЙ ОБЗОР

ВЛИЯНИЕ ПОКОЛЕНИЙ НОРМ ТЕХНОЛОГИЧЕСКИХ
ПРОЦЕССОВ НА ТЕМПЫ ДЕГРАДАЦИИ СЛОЖНЫХ
КРЕМНИЕВЫХ ИНТЕГРАЛЬНЫХ СХЕМ

Белкин Е. Н. 13

СТАТЬИ

ОБЗОР ИССЛЕДОВАНИЙ ПО ПРОБЛЕМЕ ОЦЕНКИ
ПРОИЗВОДСТВЕННО-ТЕХНОЛОГИЧЕСКИХ РИСКОВ
НЕФТЯНОГО МЕСТОРОЖДЕНИЯ

Белоусов А. С., Астапов В. Н., Мельникова Д. А. 20

ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА
SIMINTESCH ДЛЯ ОПТИМИЗАЦИИ ПЕРЕРАСПРЕДЕЛЕНИЯ
ЗАГРУЗКИ ГАЗОТУРБИННЫХ ДВИГАТЕЛЕЙ

Ильичев В. Ю. 26

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПОНЕНТА
УПРАВЛЕНИЯ ПОЛИТИКАМИ КОНСИСТЕНТНОСТИ ДАННЫХ
В ГЕОРАСПРЕДЕЛЕННОЙ БИЗНЕС-ИНФОРМАЦИОННОЙ СИСТЕМЕ

Синявина А. П., Макушин А. В. 32

CONTENTS

Technical sciences

ARTICLE

ANTIFROD SYSTEM: DEVELOPMENT OF A SOFTWARE
COMPLEX TO PREVENT FRAUDULENT TRANSACTIONS

Dubrovskiy A. A., Maleshina L. M. 5

REVIEW

INFLUENCE OF PROCESS-NODE GENERATIONS
ON THE DEGRADATION RATES OF COMPLEX
SILICON INTEGRATED CIRCUITS

Belkin E. N. 13

ARTICLES

REVIEW OF RESEARCH ON THE PROBLEM OF ASSESSING
PRODUCTION AND TECHNOLOGICAL RISKS OF AN OIL FIELD

Belousov A. S., Astapov V. N., Melnikova D. A. 20

APPLICATION OF THE SIMINTECH GENETIC ALGORITHM
FOR OPTIMIZING LOAD REDISTRIBUTION AMONG
GAS TURBINE ENGINES

Ilichev V. Y. 26

DESIGN OF A SOFTWARE COMPONENT FOR MANAGING
DATA CONSISTENCY POLICIES IN A GEO-DISTRIBUTED
BUSINESS INFORMATION SYSTEM

Sinyavina A. P., Makushin A. V. 32

СТАТЬЯ

УДК 004.056.5



CC BY 4.0

АНТИФРОД-СИСТЕМА: РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ПРЕДОТВРАЩЕНИЯ МОШЕННИЧЕСКИХ ТРАНЗАКЦИЙ

Дубровский А. А., Малёшина Л. М. ORCID ID 0000-0002-4876-462X

*Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет транспорта» (МИИТ), Москва, Российская Федерация,
e-mail: docentmlm@mail.ru*

Предотвращение мошенничества в онлайн-платежах является сложным процессом, но он имеет решающее значение для успеха компаний по всему миру. Внедрение антифрод-систем позволяет выявлять угрозы на разных уровнях, снижать риски и делать транзакции более безопасными. В России такие системы получили широкое распространение благодаря усилению регуляторных требований и активному развитию онлайн-платежей. Цель исследования – создать систему, состоящую из нескольких взаимодействующих сервисов, для автоматической проверки транзакций на возможные признаки мошенничества. В ходе разработки системы была построена архитектура, состоящая из нескольких взаимодействующих сервисов, каждый из которых выполняет конкретную задачу в рамках общей системы. Были использованы такие технологии, как gRPC для обмена сообщениями между сервисами и RabbitMQ для асинхронной обработки транзакций. Валидация транзакций и проверки на мошенничество реализованы на gRPC сервере с использованием собственной логики валидации и интеграции с базой данных, обеспечивающей хранение черного списка пользователей. Главный сервис системы предоставляет пользователям два ключевых интерфейса для взаимодействия с системой: REST API с поддержкой HATEOAS и GraphQL API. REST API с HATEOAS обеспечивает гибкость и удобство, позволяя пользователям получать не только данные о транзакциях и результатах проверок, но и ссылки на возможные дальнейшие действия, что упрощает навигацию и интеграцию с другими сервисами.

Ключевые слова: онлайн-транзакции, мошенничество в онлайн-платежах, защита данных, антифрод-система

ANTIFROD SYSTEM: DEVELOPMENT OF A SOFTWARE COMPLEX TO PREVENT FRAUDULENT TRANSACTIONS

Dubrovskiy A. A., Maleshina L. M. ORCID ID 0000-0002-4876-462X

*Federal State Autonomous Educational Institution of Higher Education
“Russian University of Transport (MIIT)”, Moscow, Russian Federation,
e-mail: docentmlm@mail.ru*

Preventing online payment fraud is a complex process, yet it is crucial to the success of companies worldwide. Implementing anti-fraud systems allows for the detection of threats at multiple levels, mitigating risks, and making transactions more secure. In Russia, such systems have become widespread due to tightened regulatory requirements and the rapid growth of online payments. The aim of the research is to create a system consisting of several interacting services to automatically check transactions for possible signs of fraud. During the system's development, an architecture was built consisting of several interacting services, each performing a specific task within the overall system. Technologies such as gRPC for messaging between services and RabbitMQ for asynchronous transaction processing were used. Transaction validation and fraud checks were implemented on the gRPC server using custom validation logic and integration with a database that stores a user blacklist. The system's main service provides users with two key interfaces for interacting with the system: a REST API with HATEOAS support and a GraphQL API. The REST API with HATEOAS provides flexibility and convenience, allowing users to access not only transaction data and audit results but also links to possible further actions, simplifying navigation and integration with other services.

Keywords: online transactions, online payment fraud, data protection, anti-fraud system

Введение

«Мошенничество в онлайн-платежах стало одной из главных сложностей для компаний, которые работают в сфере финансовых услуг. Потери средств от мошенничества могут исчисляться в миллиардах долларов, и с каждым годом способы мошенников становятся все более сложными» [1]. Мошенники продолжают придумывать все более изощренные способы

обмана, используя социальную инженерию, психологическое давление и современные технологии [2].

Антифрод-системы (от англ. anti-fraud – борьба с мошенничеством) представляют собой программные комплексы для предотвращения мошеннических транзакций. Антифрод-система обнаруживает и идентифицирует банковскую операцию, присваивая ей определенную метку в зависимости

от степени риска. И далее принимает решение: пропустить платеж, запросить дополнительную проверку или заблокировать.

Борьба с мошенничеством – это совместная работа технологий и бдительности людей. Важно не только внедрить систему, но и поддерживать ее в актуальном состоянии, а также ответственно относиться к своей цифровой безопасности. Антифрод стал неотъемлемой частью финансовой экосистемы. Он позволяет компаниям защищать клиентов и минимизировать их потери. Благодаря постоянному развитию и обучению, такие системы остаются ключевым инструментом противодействия мошенникам [3, 4].

Цель исследования – создать систему, состоящую из нескольких взаимодействующих сервисов, для автоматической проверки транзакций на возможные признаки мошенничества. Система должна предоставлять конечным пользователям интерфейсы для взаимодействия, обеспечивать высокую производительность и масштабируемость, а также поддерживать актуальность данных в реальном времени.

Материалы и методы исследования

В результате исследования было выявлено, что антифрод-система должна включать четыре основных компонента: (1) главный сервис: предоставляет REST и GraphQL API [5] для взаимодействия с пользователем; отвечает за создание и обновление транзакций; отправляет сообщения о создании транзакций в очередь RabbitMQ [6, 7]; читает сообщения из RabbitMQ для обновления статуса транзакций на основании решения от других компонентов системы; (2) gRPC клиент [8]: получает сообщения о транзакциях из RabbitMQ; обращается к gRPC серверу для вынесения вердикта о подтверждении или отклонении транзакции; передает результат обработки обратно в очередь RabbitMQ; (3) gRPC сервер [8]: реализует логику проверки транзакций на основе заданных правил и данных; возвращает результат проверки в виде решения: подтверждена или отклонена транзакция; (4) Notifier сервис: получает сообщения из RabbitMQ об обновлении статуса транзакций; пересылает обновления статусов через WebSocket [9] для обеспечения отображения актуальной информации в реальном времени на стороне клиента.

Функциональные требования к системе: пользователь должен иметь возможность создавать транзакции через REST/GraphQL API; система должна проверять каждую транзакцию на признаки мошенничества; результат проверки транзакции должен

отображаться в реальном времени с использованием WebSocket; все компоненты системы должны быть модульными и легко масштабируемыми.

Нефункциональные требования: использование RabbitMQ для обмена сообщениями между сервисами; высокая отказоустойчивость за счет распределенной архитектуры; логирование всех событий для последующего анализа и отладки; время обработки транзакции не должно превышать 2 с.

Технологический стек: Backend: Kotlin [10], Spring Boot [11]; API: REST, GraphQL; сообщения: RabbitMQ; логика проверки: gRPC; обновления в реальном времени: WebSocket.

Проектирование безопасного API требует комплексного подхода, включающего надежную аутентификацию (OAuth 2.0, API-ключи), шифрование (HTTPS, TLS), лимиты запросов (Rate Limiting), защиту от атак и постоянный мониторинг.

Результаты исследования и их обсуждение

Для реализации данного проекта был выбран подход contract-first, что подразумевает создание API-контракта на начальном этапе разработки. В рамках данного подхода была разработана отдельная библиотека, которая содержит четко определенный контракт API, а также описания основных сущностей системы. Этот контракт был документирован с использованием таких инструментов, как Swagger и OpenAPI, что позволило получить интерактивную веб-документацию (рис. 1). Такой подход позволяет добиться согласованности между различными компонентами системы и упрощает последующую разработку, тестирование и поддержку.

Данный обработчик отрабатывает только на исключения типа `ResourceNotFoundException`. Предоставляет пользователю API значащий статус код и сообщение с подсказкой о причине ошибки. Для API в контракте определены два контроллера: `user-controller` и `transaction-controller`.

Разработка GraphQL API. Была описана схема запросов `graphql`, содержащая описания типов для взаимодействия с GraphQL API. Существуют два основных типа операций: запросы (Query) и мутации (Mutation). Запросы позволяют получать данные о пользователях. Например, с помощью запроса `user` можно получить информацию об одном пользователе по его идентификатору, включая транзакции и данные учетной записи [12, 13]. Запрос `users` возвращает список всех пользователей с их данными.

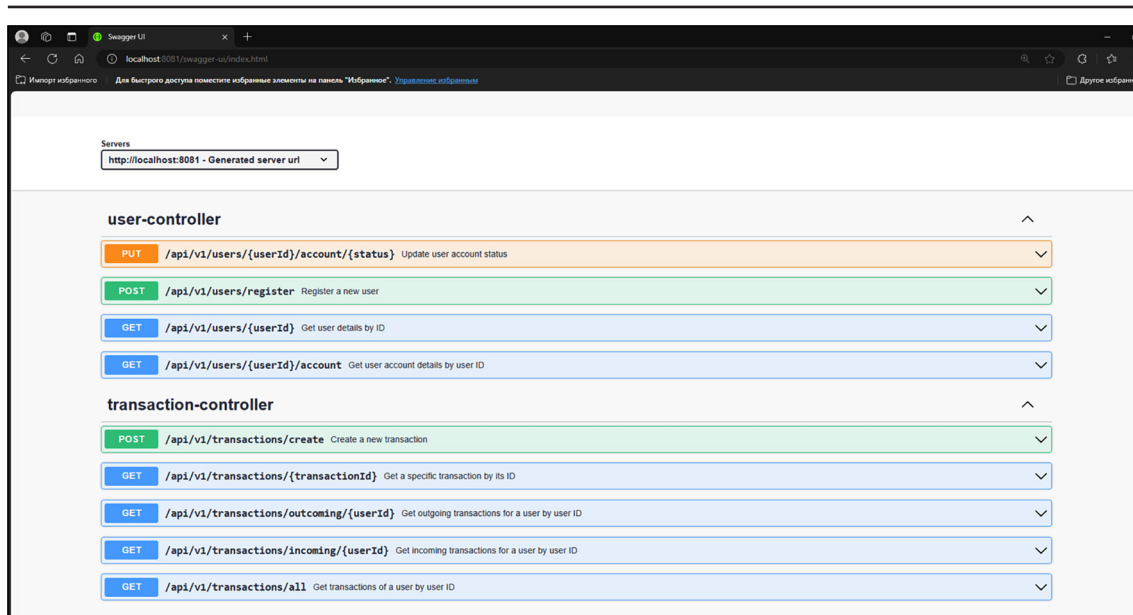


Рис. 1. Документация конечных точек API

Примечание: составлен авторами по результатам данного исследования

Отметим, что библиотека определяет общие классы ошибок и handler'ы для их обработки. Пример обработчика ошибок:

```
@ExceptionHandler(ResourceNotFoundException::class)
@ResponseStatus(HttpStatus.NOT_FOUND)
@ApiResponse(
    responseCode = "404",
    description = "Resource Not Found",
    content = [Content(schema = Schema(implementation =
ExceptionResponse::class))]
)
fun handleResourceNotFoundException(ex: ResourceNotFoundException): ResponseEntity<ExceptionResponse> {
    val response = ExceptionResponse(
        errorCode = "RESOURCE_NOT_FOUND",
        message = "Resource not found: ${ex.message}",
        details = "The resource with the given ID was not found in the system."
    )
    return ResponseEntity.status(HttpStatus.NOT_FOUND).body(response)
}
```

Мутации используются для внесения изменений в систему. Мутация createUser позволяет создать нового пользователя, предоставив такие данные, как имя, электронная почта, код валюты и тип учетной записи. Для работы с транзакциями предусмотрены мутации addIncomingTransaction и addOutcomingTransaction, которые добавляют входящие и исходящие транзакции для конкретного пользователя. Каждая транзакция содержит информацию о сумме, статусе, идентификаторах участников и способе оплаты.

Основной тип данных, с которым работает API, – это UserAggregate, который

объединяет информацию о пользователе, его транзакциях и учетной записи. Каждая транзакция представлена типом Transaction, включающим такие поля, как идентификатор, сумма, статус, отправитель, получатель и способ оплаты. Учетная запись пользователя описывается типом Account, который содержит баланс, статус, тип учетной записи и код валюты.

В схеме предусмотрены перечисления для обозначения фиксированных значений. Например, TransactionStatus определяет статус транзакции как «в ожидании» или «завершена», AccountStatus описывает

активность учетной записи, а AccountType указывает на тип учетной записи, который может быть сберегательным или расчет-

ным. Перечисление PaymentMethod задает способы оплаты, такие как кредитная карта или банковский перевод.

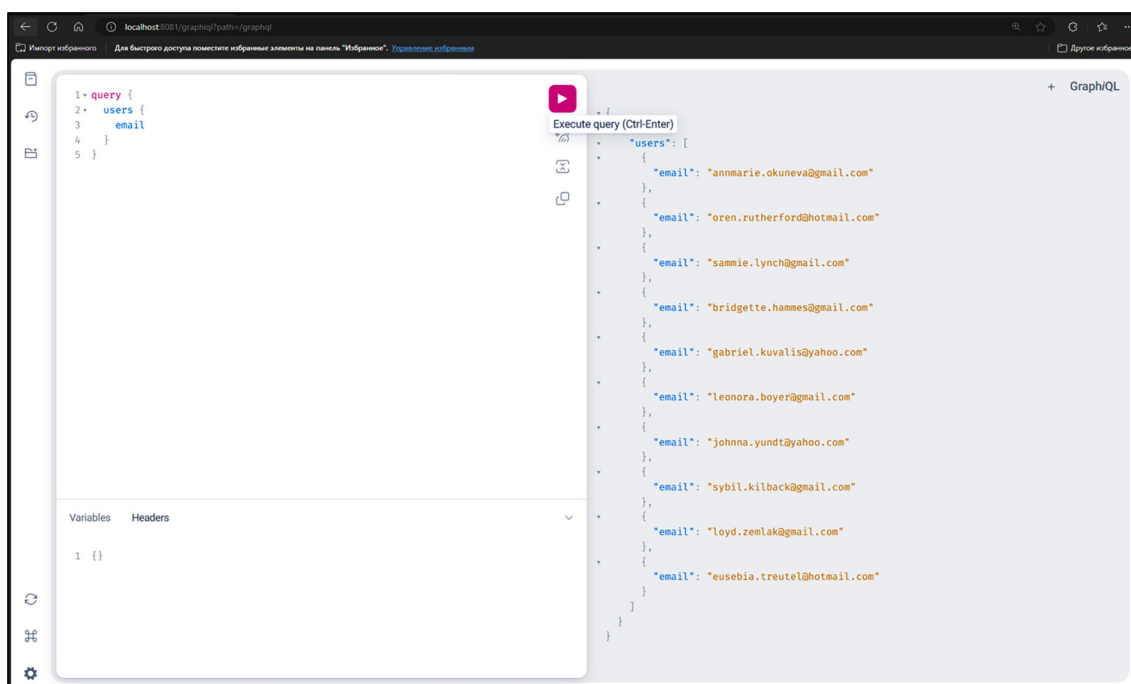


Рис. 2. Пример и результат выполнения запроса

Примечание: составлен авторами по результатам данного исследования

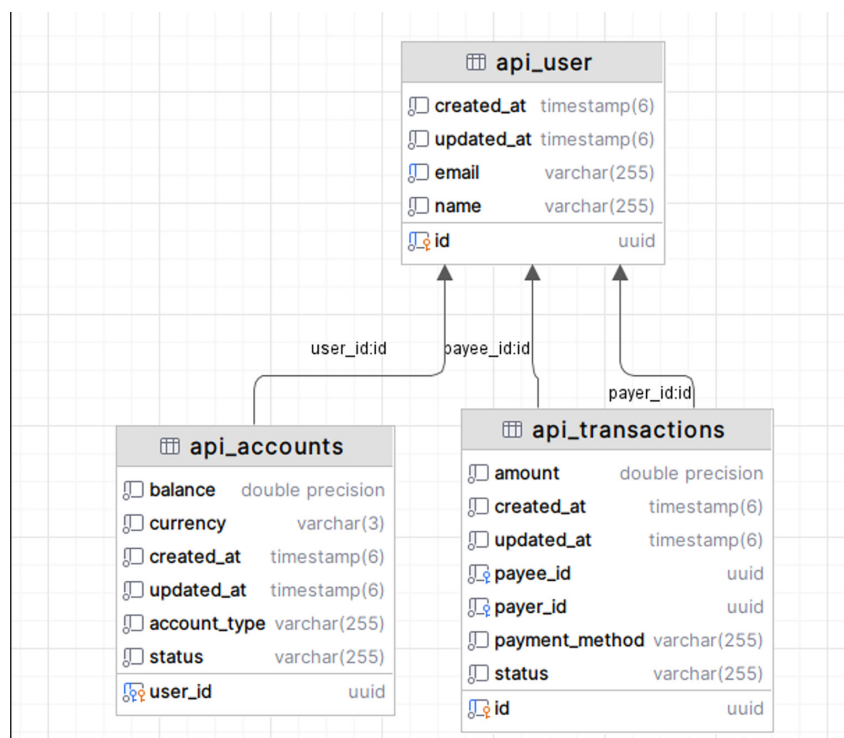


Рис. 3. Схема базы данных

Примечание: составлен авторами по результатам данного исследования

Такое API позволяет пользователю гибко запрашивать только нужные поля и в некоторой степени самому определять формат ответа от API. Как пример можно привести запрос на все e-mail'ы пользователей. Сам запрос и результат его выполнения в веб-оболочке graphiql продемонстрирован на рис. 2. Благодаря гибкой имплементации можно запросить и поля вложенной сущности.

Имплементация основного сервиса. Основной сервис разработан на базе Spring Boot, Spring Data и утилит для взаимодействия с RabbitMQ. Сервис хранит данные о пользователях в PostgreSQL [14] в схеме, представленной на рис. 3.

Как можно видеть, таблица `api_user`, представленная на рисунке 3, имеет связь

один к одному с таблицей `api_accounts`, а с `api_transactions` – один ко многим. В данном сервисе и определенном ранее API реализованы CRUD операции для этих сущностей.

Отметим, что при имплементации REST API был использован принцип HATEOAS (Hypermedia as the Engine of Application State) [15]. Все сущности имеют ссылки на связанные с ними сущности, чтобы облегчить пользователям взаимодействие с API. Наглядным примером может служить постраничный список транзакций. Запросим постраничный ответ с размером страницы равным 1 с помощью следующего url - `/api/v1/transactions/all?size=1`. Ответ будет иметь следующий формат:

```
{«_embedded»: {
  «transactionRepresentationList»: [
    { «id»: «131addb3-8686-4a7b-b431-6b3cdc2c2811»,
      «amount»: 0.8500924491606185,
      «status»: "PENDING",
      «_links»: {
        «self»: {
          «href»: "http://localhost:8081/api/v1/transactions/131addb3-8686-4a7b-b431-6b3cdc2c2811" },
        «payee»: {
          «href»: "http://localhost:8081/api/v1/users/3be2f90b-d4ac-4102-889e-872ccf7326e6" },
        «payer»: {
          «href»: "http://localhost:8081/api/v1/users/243d5f2e-32ca-40bb-a41c-e0998fd4eb3a" }}}},
    «_links»: {
      «self»: {
        «href»: "http://localhost:8081/api/v1/transactions/all?page=0&size=1" },
      «next»: {
        «href»: "http://localhost:8081/api/v1/transactions/all?page=1&size=1" }},
    «page»: {
      «size»: 1,
      «totalElements»: 10,
      «totalPages»: 10,
      «number»: 0 }}
```

Можно видеть, что ответ содержит ссылки на следующую страницу, `page` и `payee`. Данное расширение ответа позволяет значительно упростить работу с API.

Также в проект была добавлена интеграция с очередью сообщений RabbitMQ. Данный сервис будет отправлять уведом-

ления в очередь о создании новых сообщений, а также потреблять сообщения об обновлении статуса транзакций. Взаимодействие с RabbitMQ осуществляется с помощью RabbitTemplate.

Для отправки сообщений реализован `RMQTransactionSenderService`:

```
@Service
class RMQTransactionSenderService(private val amqpTemplate:
AmqpTemplate) : TransactionSenderService {
  override fun sendTransaction(transaction: TransactionMessageDTO) {
    amqpTemplate.convertAndSend(RMQConstants.Transactions.TRANSACTION_
EXCHANGE_NAME, RMQConstants.Transactions.ROUTING_KEY, transaction)
  }
}
```

Данный сервис используется непосредственно в JpaTransactionService, метод sendTransaction вызывается непосредственно при создании транзакции.

Для обновления статуса транзакции используется TransactionStatusUpdateListener:

```
@Service
class TransactionStatusUpdateListener(
    private val transactionService: TransactionService
) {
    @RabbitListener(queues = ["update-queue"])
    fun updateTransactionStatus(transactionCheckDto: TransactionCheckResultDto) {
        transactionService.updateTransactionStatus(UUID.
fromString(transactionCheckDto.transactionId), transactionCheckDto.
status)
    }
}
```

При получении сообщения об изменении статуса транзакции слушатель обращается к transactionService, который и обновляет значение статуса в базе данных.

Имплементация gRPC клиента и сервера. В данной архитектуре gRPC клиент будет отвечать за отправку и получение сообщений через RabbitMQ, обеспечивая надежную и асинхронную коммуникацию между различными сервисами [16]. Основная логика валидации данных, выполнение бизнес-правил и условий, будет сосредоточена на gRPC сервере. Важно отметить, что как клиент, так и сервер будут реализованы без использования Spring Boot, что дает большую гибкость в выборе технологий и уменьшает зависимость от сторонних фреймворков. Такой подход обеспечит высокую производительность и расширяемость системы, сохраняя при этом простоту в реализации и поддержку.

В первую очередь также была реализована библиотека с gRPC контрактом на основе protobuf. Проект состоит из .proto файла с объявлением сообщений и gRPC сервисов.

Стоит отметить, что proto файл настроен таким образом, чтобы сгенерированные классы находились com.example.antifraud пакете и были доступны из внешнего класса AntiFraudService. Структура сгенерированных классов представлена на рис. 4.

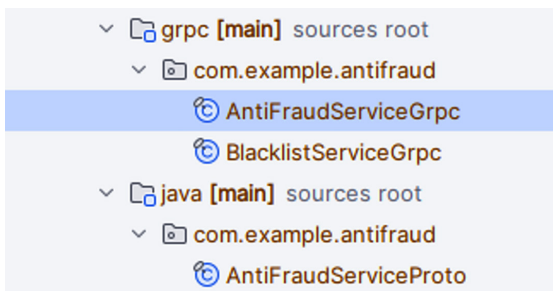


Рис. 4. Структура сгенерированных классов
Примечание: составлен авторами по результатам данного исследования

Далее был реализован gRPC клиент без использования Spring Boot. Для удобства работы с общими значениями в проекте был создан объект Constants в пакете constants, который содержит константы, используемые по всему клиенту. Основная логика клиента реализована в классе grpcClient, который предоставляет базовую имплементацию для общения с сервером через сгенерированные stub классы. Этот клиент служит посредником между приложением и сервером, обеспечивая корректную отправку запросов и получение ответов. В файле Main.kt объявляются необходимые очереди с использованием стандартного клиента RabbitMQ, после чего регистрируется callback, который будет вызываться при поступлении сообщения в очередь TRANSACTION_CREATION_QUEUE. Внутри этого callback выполняется вызов gRPC сервера, что обеспечивает взаимодействие между RabbitMQ и gRPC сервисом для дальнейшей обработки сообщений. Такой подход позволяет легко интегрировать асинхронные очереди с сервисом. Структура проекта представлена на рис. 5.

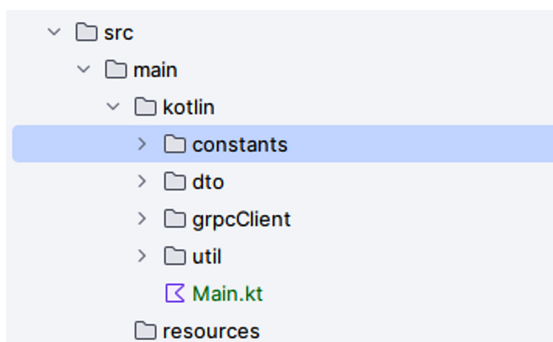


Рис. 5. Структура gRPC-клиента
Примечание: составлен авторами по результатам данного исследования

gRPC сервер, в свою очередь, также не использует Spring Boot, поскольку сам

по себе `io.grpc.Server` является HTTP сервером. Для реализации gRPC сервера применяется веб-сервер `Netty`, который является стандартной имплементацией для gRPC. Сервер взаимодействует с базой данных, которая используется для ведения черного списка пользователей. Для работы с базой данных применяется библиотека `Ktorm`, которая обеспечивает удобное взаимодействие с базой данных и выполняет операции с объектами, используя функциональные подходы `Kotlin`. В качестве базы данных используется in-memory база данных `H2`.

Пакет, отвечающий за работу с базой данных, включает в себя объявление объектов, которые обеспечивают доступ к данным, а также функции для их инициализации и выполнения CRUD операций. Для валидации транзакций реализованы различные условия, которые включают в себя проверку размера переводимых средств в контексте выбранного способа перевода, проверку наличия одного из пользователей в черном списке и другие необходимые проверки.

Имплементация `Notifier-service`. `Notifier-service` является Spring Boot приложением, которое использует стандартные механизмы работы с WebSocket, предоставляемые Spring. Основная цель сервиса – обеспечить асинхронное уведомление пользователей через WebSocket при получении сообщений из RabbitMQ. Проект включает несколько ключевых компонентов:

1. конфигурация очередей RabbitMQ – определена очередь для приема сообщений, где реализован класс для парсинга сообщений. Этот класс отвечает за корректную обработку данных, поступающих из RabbitMQ, перед их отправкой на WebSocket;

2. конфигурация WebSocket – настроен WebSocket сервер с определением endpoint-a и топика, который будет использоваться для отправки сообщений клиентам;

3. `NotifierService` – основной сервис, который слушает очередь RabbitMQ, получает сообщения и перенаправляет их через WebSocket на подключенные клиентские приложения. Этот сервис выполняет роль посредника, получая данные из очереди и обеспечивая их доставку в нужное место.

Для демонстрации работы WebSocket был создан статический ресурс `index.html`, который включает вложенный JavaScript скрипт. Этот скрипт позволяет устанавливать соединение с сервером через WebSocket и в реальном времени обновлять содержимое HTML страницы.

Скрипт использует библиотеки `SockJS` и `STOMP.js` для организации взаимодействия с сервером. После подключения

к WebSocket серверу клиент подписывается на определенный топик и начинает получать сообщения, связанные с транзакциями. Каждый раз, когда сервер отправляет новое сообщение, скрипт обрабатывает его и обновляет соответствующий элемент на странице, отображая информацию о транзакции, такую как ее идентификатор, статус и описание. Это позволяет пользователю наблюдать за происходящими событиями в реальном времени без необходимости вручную обновлять страницу.

Заключение

В ходе разработки системы для автоматической проверки транзакций на возможные признаки мошенничества была построена архитектура, состоящая из нескольких взаимодействующих сервисов, каждый из которых выполняет конкретную задачу в рамках общей системы. Были использованы такие технологии, как gRPC для обмена сообщениями между сервисами и RabbitMQ для асинхронной обработки транзакций. Валидация транзакций и проверки на мошенничество реализованы на gRPC сервере с использованием собственной логики валидации и интеграции с базой данных, обеспечивающей хранение черного списка пользователей.

Главный сервис системы предоставляет пользователям два ключевых интерфейса для взаимодействия с системой: REST API с поддержкой HATEOAS и GraphQL API. REST API с HATEOAS обеспечивает гибкость и удобство, позволяя пользователям получать не только данные о транзакциях и результатах проверок, но и ссылки на возможные дальнейшие действия, что упрощает навигацию и интеграцию с другими сервисами. В свою очередь, GraphQL API позволяет пользователям запрашивать только необходимые данные.

Для уведомлений пользователей о результатах проверки транзакций был создан `Notifier-service` на Spring Boot, который использует RabbitMQ и WebSocket для отправки уведомлений в реальном времени. Это решение позволило пользователям получать актуальные данные без задержек, что важно для эффективной работы с системами, основанными на обработке финансовых транзакций.

Разработан прототип архитектуры системы, которая обеспечивает автоматическую проверку транзакций на возможные признаки мошенничества и предоставляет интерфейсы для взаимодействия с конечными пользователями. Эффективность антифрод-проверок требует экспериментальной валидации.

Для обеспечения безопасности архитектуры необходимо применять комплексный подход.

Список литературы

1. Васильев Т. И. Разработка и внедрение систем для борьбы с мошенничеством в финансовых транзакциях // Экономика. Информатика. 2024. Т. 51. № 4. С. 919–925. URL: <https://econom-inform-journal.ru/index.php/journal/article/view/408> (дата обращения: 26.06.2026). DOI: 10.52575/2687-0932-2024-51-4-919-925.
2. Брехов Д. А. Способы совершения мошеннических действий с использованием сети Интернет, средств подвижной связи и систем дистанционного банковского обслуживания // Вестник Поволжского института управления. 2021. Т. 21. № 4. С. 52–60. URL: <https://vestnik-piu.ru/arkhiv/2021/4/Brekhov.pdf> (дата обращения: 26.06.2026). DOI: 10.22394/1682-2358-2021-4-52-60.
3. Сенцова А. Ю., Тимергазин В. Э., Ильясова Р. И. Антифрод-система как инструмент предотвращения мошенничества // Информационные технологии. Проблемы и решения. 2021. № 4 (17). С. 101–107. EDN: JMFOVP.
4. Копнин А. А., Соколова Е. В., Долгополов А. А. Методика обеспечения безопасности банковских интернет-транзакций на основе антифрод системы // International Journal of Professional Science. 2022. № 10. С. 149–157. URL: <https://cyberleninka.ru/article/n/metodika-obespecheniya-bezopasnosti-bankovskih-internet-tranzaktsiy-na-osnove-antifrod-sistemy/viewer> (дата обращения: 26.06.2026). DOI: 10.54092/25421085_2022_10_149.
5. Бэнкс А., Порселло Е. GraphQL. Язык запросов для современных веб-приложений. СПб.: Питер, 2019. 240 с. (Серия «Бестселлеры O'Reilly»). ISBN 978-5-4461-1143-5.
6. Videla A., Williams J. J. W. RabbitMQ in Action / 1st Edition. Manning Publishing. 2012. 312 p. ISBN 9781935182979.
7. Официальная документация RabbitMQ. [Электронный ресурс]. URL: <https://www.rabbitmq.com/documentation.html> (дата обращения: 26.06.2026).
8. Индрасири К., Куруппу Д. gRPC: запуск и эксплуатация облачных приложений. Go и Java для Docker и Kubernetes. СПб.: Питер, 2021. 224 с. (Серия «Бестселлеры O'Reilly»). ISBN 978-5-4461-1737-6.
9. Хабаров С. П., Шилкина М. Л. Построение распределенных систем на базе WebSocket: учебное пособие для вузов. 3-е изд., стер. СПб.: Лань, 2026. 216 с. ISBN 978-5-507-54698-5.
10. Сошин А. Kotlin. Паттерны проектирования и лучшие практики. 3-е изд. Sprint book. 2025. 416 с. (Серия «Библиотека программиста»). ISBN 978-601-09-9694-6.
11. Хеклер М. Spring Boot по-быстрому. СПб.: Питер, 2022. 352 с. (Серия «Бестселлеры O'Reilly»). ISBN 978-5-4461-3942-2.
12. Гумерова Г. Р., Мансурова Т. Г. Моделирование требований к программному обеспечению // Вестник Алтайской академии экономики и права. 2023. № 12–1. С. 42–52. URL: <https://vael.ru/ru/article/view?id=3131> (дата обращения: 26.06.2026). DOI: 10.17513/vael.3131.
13. Патент № 2388053 С1 Российская Федерация, МПК G06Q 20/00. Способ проверки транзакций, автоматическая система для проверки транзакций и узел для проверки транзакций (варианты): № 2008143735/09; заявл. 06.11.2008; опубл. 27.04.2010 / А. Г. Рожков.
14. Жао Э. SQL. Pocket guide. 4-е изд. Астана: Спринт Бук, 2024. 320 с. ISBN 978-601-08-3728-7.
15. Aydemir F., Basciftci F. Application of HATEOAS Principle in RESTful API Design // IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo), Budapest, Hungary. 2022. P. 000051–000056. URL: <https://ieeexplore.ieee.org/document/10029427> (дата обращения: 26.06.2026). DOI: 10.1109/CINTI-MACRo57952.2022.10029427.
16. Баженов А. Э., Райков А. В., Нехаев М. В., Ореховский Н. В. Оптимизация взаимодействия микросервисов с использованием асинхронных вызовов и брокеров сообщений (Kafka, RabbitMQ) // Программные системы и вычислительные методы. 2025. № 4. С. 77–93. URL: https://nbpublish.com/library_read_article.php?id=77228 (дата обращения: 26.06.2026). DOI: 10.7256/2454-0714.2025.4.77228.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.

НАУЧНЫЙ ОБЗОР

УДК 621.3.049.774:621.382.3:004.31



CC BY 4.0

ВЛИЯНИЕ ПОКОЛЕНИЙ НОРМ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ НА ТЕМПЫ ДЕГРАДАЦИИ СЛОЖНЫХ КРЕМНИЕВЫХ ИНТЕГРАЛЬНЫХ СХЕМ

Белкин Е. Н.

*Акционерное общество «АЛГОНТ», Калуга, Российская Федерация,
e-mail: belkinen@algont.ru*

Деградация современных кремниевых интегральных схем определяется не номинальным числом нанометров само по себе, а совместным действием электрических полей, температуры, плотности тока, материалов и профиля нагрузки. Цель работы – систематизировать сведения о том, как переход от 180 нм к номинальным 2–3 нм изменяет набор механизмов старения сложных микросхем, включая центральные и графические процессоры. Выполнен обзор публикаций 2016–2025 гг. в IEEE Xplore, Google Scholar и eLIBRARY.RU; Crossref использован для проверки DOI, дополнительно просмотрены списки литературы. Из 47 выявленных публикаций после отбора включены 26 современных и 4 фундаментальные работы. Для сопоставления использованы модели Аррениуса, Блэка, температурной нестабильности смещения и пробоя диэлектрика во времени. Значимость механизмов по группам техпроцессов оценивалась по формализованному кодированию литературных выводов по шкале 1–5. Показано, что для 180–130 нм в литературе чаще выделяются горячие носители и износ оксида; для 90–28 нм – температурная нестабильность смещения, пробой диэлектрика и надежность медной металлизации; для 22–2 нм могут становиться критичными самонагрев, контакты, средние и верхние уровни межсоединений и стохастические дефекты. Для процессоров скорость старения существенно зависит от тепловой карты и распределения активности, поэтому применяются динамические запасы, мониторы старения и управление ресурсами во время работы. Сделан вывод, что число нанометров не позволяет рассчитать срок службы конкретного изделия без калиброванных параметров технологии и эксплуатационного профиля.

Ключевые слова: кремниевая интегральная схема, техпроцесс, деградация, надежность, электромиграция, самонагрев, старение, процессор

INFLUENCE OF PROCESS-NODE GENERATIONS ON THE DEGRADATION RATES OF COMPLEX SILICON INTEGRATED CIRCUITS

Belkin E. N.

ALGONT Joint Stock Company, Kaluga, Russian Federation, e-mail: belkinen@algont.ru

Degradation of modern silicon integrated circuits is determined not by the nominal nanometer number itself, but by the combined action of electric fields, temperature, current density, materials, and workload profile. The purpose of this paper is to systematize how the transition from 180 nm to nominal 2–3 nm changes the set of aging mechanisms in complex chips, including central and graphics processors. Publications from 2016–2025 were reviewed in IEEE Xplore, Google Scholar, and eLIBRARY.RU; Crossref was used to verify DOI data, and reference lists were also screened. Of 47 identified publications, 26 recent and 4 foundational papers were included after title, abstract, and full-text screening. Arrhenius, Black, bias temperature instability, and time-dependent dielectric breakdown models were used for comparison. The significance of mechanisms across process-node groups was assessed by formalized coding of literature conclusions on a scale from 1 to 5. The literature more often highlights hot carriers and oxide wear for 180–130 nm; bias temperature instability, dielectric breakdown, and copper-interconnect reliability for 90–28 nm; and self-heating, contacts, middle and upper interconnect levels, and stochastic defects as potentially critical for 22–2 nm. In processors, aging rate depends strongly on the thermal map and activity distribution, which motivates dynamic margins, aging monitors, and runtime resource management. The nominal nanometer number therefore cannot be used to calculate the lifetime of a specific product without process-calibrated parameters and an operating profile.

Keywords: silicon integrated circuit, process node, degradation, reliability, electromigration, self-heating, aging, processor

Введение

Под деградацией в работе понимается накопленное изменение параметров транзисторов, межсоединений и корпуса, приводящее к росту задержек, утечек, сопротивлений, шумов и вероятности сбоев. Для цифровой логики основным наблюдаемым результатом становится уменьше-

ние временных запасов, для памяти – рост ошибок чтения и записи, для аналоговых узлов – смещение порогов, коэффициентов усиления и рабочих точек.

Название технологического поколения – 180 нм, 65 нм, 7 нм или 2 нм – не является самостоятельной физической причиной отказа. Оно лишь коррелирует с длиной кана-

ла, эквивалентной толщиной диэлектрика, рабочим напряжением, материалами затвора и межсоединений, плотностью мощности и тепловым сопротивлением. После завершения классического масштабирования Дэннарда напряжение питания снижалось медленнее геометрических размеров, поэтому электрические поля и плотность мощности не уменьшались пропорционально [1].

В сложных микросхемах деградация распределена неоднородно. Кешы, регистровые файлы, арифметические блоки, сети питания, регуляторы и корпус имеют разные карты активности и температуры. Часть рисков компенсируется проектными запасами, коррекцией ошибок, терморегулированием, встроенными датчиками и статическим временным анализом с учетом старения [2; 3]. Поэтому сравнение поколений техпроцессов должно отделять подтвержденные физические зависимости от качественных обобщений по разнородной литературе.

Цель исследования – обобщить, как смена поколений технологических норм влияет на механизмы деградации сложных кремниевых интегральных схем, формализовать литературное сопоставление рисков и определить факторы, которые важнее номинального числа нанометров при оценке надежности.

Материалы и методы исследования

Работа выполнена как обзор с элементами формализованного картирования литературы. Основной временной интервал составлял 2016–2025 гг.; более ранние публикации включались только как фундаментальные источники моделей или механизмов. Основной поиск проводился в IEEE Xplore, Google Scholar и eLIBRARY.RU, а Crossref использовался для проверки DOI и библиографических данных. Дополнительные работы выявлялись по спискам литературы. Использовались запросы: “CMOS aging process node”, “FinFET GAUFET reliability self-heating”, “electromigration BEOL sub-10 nm”, “runtime aging management processor”, “aging monitor guardband”, “workload-dependent aging CPU GPU”, а также русскоязычные сочетания «деградация КМОП техпроцесс», «электромиграция суб-10 нм» и «старение процессоров нагрузка».

По итогам целевого поиска и просмотра списков литературы сформирована рабочая выборка из 47 публикаций. После оценки заголовков, аннотаций и полного текста включены 26 работ 2016–2025 гг. и 4 фундаментальные работы; 17 публикаций исключены из-за несоответствия объекту исследования, отсутствия анализа долговременного старения, дублирования выводов

либо недостаточных библиографических данных. Критериями включения были рецензируемый статус, наличие модели, измерений или схемотехнического анализа и явная связь с температурой, напряжением, плотностью тока, поколением технологии либо профилем нагрузки. Исключались маркетинговые материалы, форумные сообщения, работы только о радиационных мягких ошибках и публикации без проверяемого первоисточника. Российская работа по оценке 65-нм процесса использована как пример комплексного учета пробоя диэлектрика, горячих носителей, температурной нестабильности смещения, электромиграции и переходных отверстий [4].

Для формализации таблицы и рисунка 3 использованы 19 публикаций с явной привязкой к поколениям проектных норм; остальные 11 – для анализа управления старением, защитных запасов, встроенных мониторов и зависимости надежности от нагрузки и тепловой карты. Каждая из 19 публикаций кодировалась по четырем группам проектных норм и пяти механизмам. Вес 0 означал отсутствие вывода для данной пары «группа – механизм», 1 – упоминание как существенного или вторичного фактора, 2 – указание как одного из основных ограничений. Нормированный показатель рассчитывался как

$$d = \Sigma w / (2N),$$

где N – число публикаций, рассматривавших группу. Значение переводилось в индекс: 1 при $d \leq 0,20$; 2 при $0,20 < d \leq 0,40$; 3 при $0,40 < d \leq 0,60$; 4 при $0,60 < d \leq 0,80$; 5 при $d > 0,80$. «Чаще выделяемыми» в таблице названы механизмы с индексом не ниже 4. Одна публикация могла учитываться в нескольких группах, поэтому значения N не суммируются.

Результаты исследования и их обсуждение

Базовые модели и границы применимости

Температура ускоряет диффузию вакансий, разрыв химических связей, генерацию ловушек, релаксацию механических напряжений и рост сопротивлений. Для перехода от температуры ускоренного испытания T_{stress} к рабочей температуре T_{use} применяют коэффициент Аррениуса:

$$AF_T = \exp \left[\frac{E_a}{k_B} \cdot \left(\frac{1}{T_{use}} - \frac{1}{T_{stress}} \right) \right], \quad (1)$$

где E_a – энергия активации,

$$k_B = 8,617 \cdot 10^{-5} \text{ эВ/К}.$$

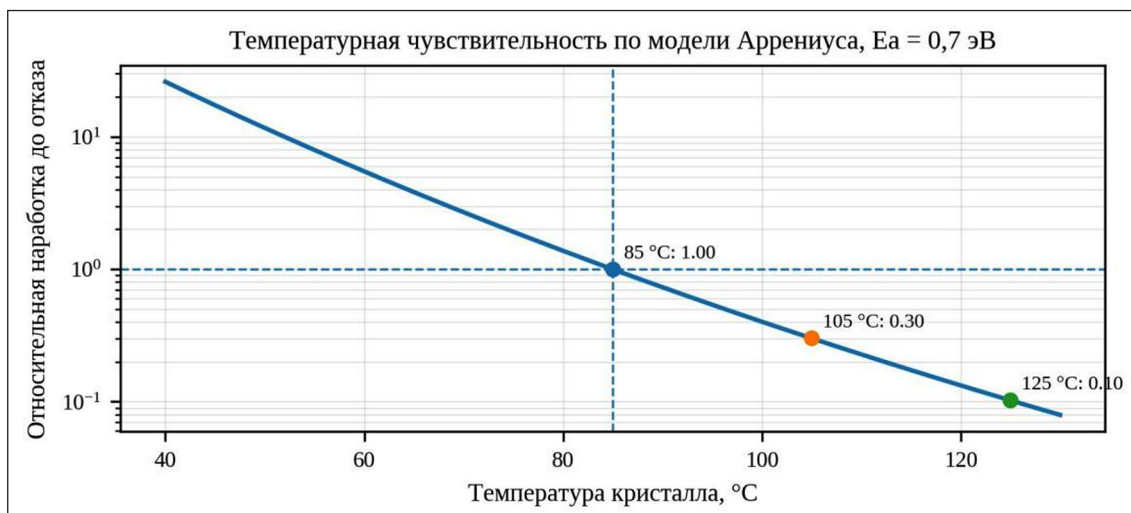


Рис. 1. Относительная средняя наработка до отказа при изменении температуры кристалла
Примечание: составлено автором по модельному расчету Аррениуса
при $E_a = 0,7$ эВ и базовой точке 85°C

В иллюстративном расчете при $E_a = 0,7$ эВ и неизменной плотности тока повышение температуры с 85 до 105°C уменьшает относительную среднюю наработку до отказа (mean time to failure, MTTF) примерно до $0,30$, а до 125°C – до $0,10$ (рис. 1).

Электромиграция описывается уравнением Блэка [5]:

$$MTTF_{EM} = A \cdot J^{-n} \cdot \exp\left(\frac{E_a}{k_B \cdot T}\right), \quad (2)$$

где J – локальная плотность тока, n – показатель токового ускорения, A – коэффициент, зависящий от геометрии, микроструктуры

и материалов. При уменьшении ширины линий, переходных отверстий и контактов локальная J может возрастать даже при снижении общего тока, поэтому в современных технологиях внимание смещается к сети питания, контактам и многоуровневой металлизации [6; 7].

Для температурной нестабильности смещения и части механизмов старения диэлектрика используют степенную зависимость от времени и напряжения [8]:

$$\Delta V_{th}(t) \approx K \cdot V^m \cdot \exp\left(-\frac{E_a}{k_B \cdot T}\right) \cdot t^n, \quad (3)$$

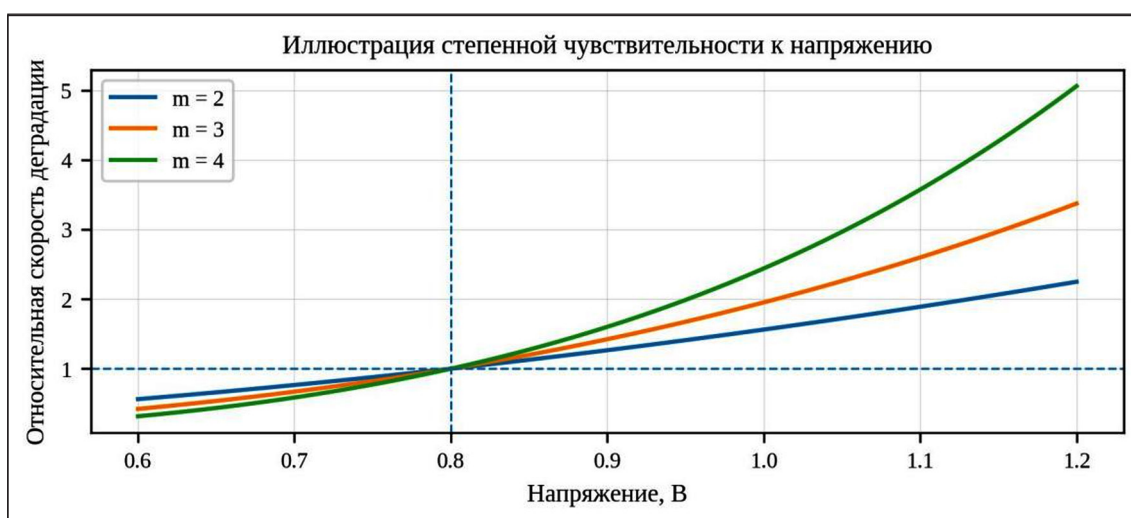


Рис. 2. Модельная чувствительность скорости деградации к напряжению при степенном законе
Примечание: составлено автором; зависимость не является паспортной характеристикой конкретной технологии

Сдвиг порогового напряжения увеличивает задержки и меняет токи утечки. При показательном значении $m = 3$ повышение напряжения с 0,8 до 1,0 В увеличивает модельную скорость деградации в 1,95 раза, а до 1,1 В – в 2,60 раза. Иллюстрация чувствительности приведена на рис. 2.

Для пробоя диэлектрика во времени применяются экспоненциальные или степенные модели поля [9]:

$$t_{BD} \approx A \cdot \exp(\gamma / E_{ox}) \text{ или } t_{BD} \approx A \cdot E_{ox}^{-p}, \quad (4)$$

Параметры E_a , n , m , p , γ , A и K не являются универсальными. Они зависят от материала диэлектрика и проводника, геометрии, формы импульсов, коэффициента заполнения, режима восстановления, диапазона температур и методики испытаний. Калибровка одного техпроцесса не переносится на другой без экспериментальной проверки; при одновременном действии самонагрева, просадок питания и нескольких механизмов простое перемножение коэффициентов ускорения также может быть неверным. Поэтому рис. 1 и 2 показывают чувствительность моделей, а не паспортный ресурс конкретного процессора.

Факторы и физические механизмы

Далее используются сокращения: HCI – деградация горячими носителями; BTI – температурная нестабильность смещения; TDDb – пробой диэлектрика во времени; EM – электромиграция; MOL и BEOL – средние и верхние уровни межсоединений. К ключевым эксплуатационным факторам относятся локальная температура, электрическое поле, плотность тока и статистическая вариабельность. В трехмерных транзисторах температура канала и контактов может превышать показания внешнего дат-

чика из-за малых теплопроводных сечений. Повышение напряжения и переходные выбросы одновременно усиливают несколько видов старения, а длительные вычислительные нагрузки увеличивают средний ток сети питания.

Горячие носители создают интерфейсные состояния и заряд в диэлектрике, что снижает ток насыщения и увеличивает задержку. Механизм особенно заметен в короткоканальных планарных и высоковольтных транзисторах, но сохраняется и в новых геометриях [10; 11]. Температурная нестабильность смещения меняет пороговое напряжение и ухудшает запасы статической памяти и критических путей [8]. Пробой диэлектрика во времени связан с накоплением ловушек и образованием проводящего пути; в структурах с охватывающим затвором возрастает роль внутренних диэлектриков и средних уровней межсоединений [9].

Электромиграция и миграция напряжений образуют пустоты и увеличивают сопротивление металлизации. Для медных и low-k систем важны барьерные и интерфейсные пути, а для субдесятинанометровых маршрутов – размерный эффект, шероховатость и контактное сопротивление [6; 7]. Самонагрев действует как общий усилитель: он способен ускорять температурную нестабильность смещения, горячие носители, пробой диэлектрика и электромиграцию одновременно [3].

Влияние поколения проектной нормы

Сравнение поколений нельзя сводить к правилу «меньше нанометров – быстрее деградация». Масштабирование снижает рабочее напряжение и улучшает электростатику, но одновременно увеличивает плотность мощности, роль контактов и чувствительность к единичным дефектам.

Формализованное сопоставление поколений проектных норм и рисков деградации

Группа и число источников	Типичные особенности	Механизмы с индексом 4–5
180–130 нм (N = 4)	Планарные структуры, более толстые оксиды, крупные линии и контакты, напряжения около 1,8–1,2 В	Горячие носители (5); износ оксида и пробой диэлектрика (4)
90–28 нм (N = 8)	Тонкие SiON/high-k диэлектрики, медь и low-k, рост утечек, медленное снижение напряжения	BTI (5); HCI (4); TDDb (4); EM и повреждения low-k (4)
22–10 нм (N = 7)	FinFET/tri-gate, трехмерная геометрия, высокая роль контактов и топологии	Самонагрев (5); BTI (4); TDDb и MOL (4); EM и межсоединения (4)
7–2 нм (N = 5)	Нанолиты и охватывающий затвор, дальнейшее уменьшение via, контактов и шага металлизации	TDDb и MOL (5); BTI (4); EM и BEOL (4); самонагрев (4)

Примечание: составлено автором по формализованному кодированию включенных публикаций; ключевые опорные источники [3; 7; 15]. N – число работ, явно рассматривавших группу; одна работа могла учитываться в нескольких группах.



Рис. 3. Индекс литературной значимости механизмов по группам проектных норм
 Примечание: составлено автором по описанной методике на основе включенных публикаций [3; 7; 15]; 1 – механизм редко выделяется, 5 – часто называется одним из основных ограничений.
 MOL – средние уровни межсоединений; BEOL – верхние уровни межсоединений
 Индекс не является измеренной скоростью отказов

Схемотехнические расчеты показывают рост чувствительности ряда параметров к старению при уменьшении минимальных размеров [12; 13], а переход к вертикально уложенным нанолентам и транзисторам с затвором вокруг канала переносит часть ограничений в контакты, внутренние диэлектрики и средние уровни межсоединений [14; 15]. Результаты формализованного обзора приведены в таблице.

Полная матрица индексов показана на рис. 3. Значения отражают частоту и силу формулировок в отобранной литературе, а не измеренную вероятность отказа и не ранжирование продукции разных фабрик.

Высокую подтвержденность имеют общие зависимости от температуры, поля и плотности тока, поскольку они следуют из физических моделей и многократно наблюдались экспериментально. Среднюю подтвержденность имеют выводы о том, какие механизмы чаще обсуждаются для групп проектных норм. Прямые сопоставления нескольких фабрик при одинаковых напряжениях, тепловых условиях и критериях отказа практически отсутствуют; поэтому утверждения о «доминировании» механизма для всего поколения техпроцессов считаются неподтвержденными и в настоящей работе не используются.

Особенности центральных и графических процессоров

В центральных и графических процессорах старение зависит от карты активности. Арифметические блоки, кеши, реги-

стровые файлы, контроллеры памяти, сети питания и корпус формируют разные тепловые карты. Поэтому изделия одной модели могут накапливать различный ущерб при офисной нагрузке, играх, рендеринге, вычислениях круглосуточно или серверной эксплуатации.

Фиксированные временные и напряженческие запасы часто выбираются по худшему сочетанию температуры и старения. В литературе рассматриваются динамический выбор запаса, прогноз по встроенным датчикам и управление ресурсами с учетом нескольких механизмов [16–18]. Для графических процессоров предложены долговременное управление надежностью, программно зависимый запас и учет зависящих от нагрузки горячих областей [19–21]. Программно управляемые датчики и мониторинг задержки уточняют потерю временного запаса [22; 23], а отображение задач учитывает тепловую карту и циклы [24; 25]. Дополнительно анализируются старение тактового дерева и совместное влияние напряжения, температуры и старения [26; 27].

Эти методы не отменяют физический износ и не дают универсального срока службы. Они позволяют сократить избыточный запас, распределить нагрузку и заранее обнаружить потерю временного запаса. Анализ критических путей с учетом нагрузки, динамический временной анализ с учетом старения и вариативности и статистическая оптимизация библиотек дополнительно связывают остаточный запас с фактической активностью схемы [28–30].

Практические рекомендации по ограничению температуры, напряжения и разгона следует рассматривать только как общие инженерные принципы; расчет ресурса конкретного центрального или графического процессора требует закрытых параметров техпроцесса, корпуса, критериев отказа и статистики партии.

Ограничения обзора. Обзор ограничен неоднородностью терминов проектных норм, различиями материалов и условий ускоренных испытаний, а также недостатком открытых данных производителей. Часть публикаций рассматривает устройства или тестовые структуры, а не полный процессор; перенос таких результатов на изделие возможен только качественно. Кодирование выполнено одним автором, поэтому сохраняется риск субъективной классификации. Шкала 1–5 снижает произвольность обобщения, но остается индексом литературы и не заменяет метаанализ измеренных времен отказа.

Выводы

1. Проектная норма влияет на старение опосредованно – через архитектуру транзистора, материалы, рабочее напряжение, геометрию контактов, плотность тока, тепловое сопротивление и разброс параметров.

2. Температура, электрическое поле и локальная плотность тока являются подтвержденными ускорителями деградации, однако численные коэффициенты применимы только после калибровки под конкретный техпроцесс и режим испытаний.

3. Для 180–130 нм в литературе чаще выделяются горячие носители и износ оксида; для 90–28 нм – температурная нестабильность смещения, пробой диэлектрика и надежность медной металлизации; для 22–2 нм могут становиться критичными самонагрев, контакты, средние и верхние уровни межсоединений и стохастические дефекты.

4. Распределение механизмов по поколениям имеет среднюю доказательность и не означает, что один механизм доминирует во всех изделиях данного номинального техпроцесса.

5. Для центральных и графических процессоров профиль нагрузки и тепловая карта могут быть не менее важны, чем поколение технологии; динамические запасы, мониторы и управление ресурсами помогают контролировать потерю временного запаса.

6. Номинальное число нанометров и общие рекомендации по температуре, напряжению или разгону не позволяют вычислить срок службы конкретного изделия без технологической калибровки и эксплуатационной статистики.

Список литературы

1. Dennard R. H., Gaensslen F. H., Yu H. N., Rideout V. L., Bassous E., LeBlanc A. R. Design of ion-implanted MOSFETs with very small physical dimensions // *IEEE Journal of Solid-State Circuits*. 1974. Vol. 9. № 5. P. 256–268. DOI: 10.1109/JSSC.1974.1050511.
2. Строгонов А. В. Методы прогнозирования долговечности ИС по параметрическим отказам // *Электроника: наука, технология, бизнес*. 2024. № 2 (233). С. 88–96. DOI: 10.22184/1992-4178.2024.233.2.88.96.
3. Chhabria V. A., Sapatnekar S. S. Impact of self-heating on performance and reliability in FinFET and GAAFET designs // *Proceedings of the 20th International Symposium on Quality Electronic Design*. 2019. P. 235–240. DOI: 10.1109/ISQED.2019.8697786.
4. Сивченко А. С., Кузнецов Е. В. Определение основных параметров надежности КМОП процесса полупроводниковой фабрики // *Наноиндустрия*. 2018. № 9 (82). С. 257–263. DOI: 10.22184/1993-8578.2018.82.257.263.
5. Black J. R. Electromigration – A brief survey and some recent results // *IEEE Transactions on Electron Devices*. 1969. Vol. 16. № 4. P. 338–347. DOI: 10.1109/T-ED.1969.16754.
6. Махвиладзе Т. М., Сарычев М. Е. Влияние точечных дефектов на возникновение электромиграции в проводнике с примесью // *Микроэлектроника*. 2021. Т. 50. № 5. С. 376–383. DOI: 10.31857/S0544126921040074.
7. Амиров И. И., Куприянов А. Н., Наумов В. В., Изюмов М. О., Волошин Д. Г., Кропоткин А. Н., Лопав Д. В., Рахимова Т. В. Основные аспекты формирования металлизации в субдесятинанометровой технологии изготовления интегральных схем // *Вестник РФФИ*. 2023. Т. 2. № 118. С. 63–76. DOI: 10.22204/2410-4639-2023-118-02-63-76.
8. Zhang J. F., Gao R., Duan M., Ji Z., Zhang W. D., Marsland J. Bias Temperature Instability of MOSFETs: Physical Processes, Models, and Prediction // *Electronics*. 2022. Vol. 11. № 9. Article 1420. DOI: 10.3390/electronics11091420.
9. Stathis J. H., La Rosa G. Reliability limits for the gate insulator in CMOS technology // *IBM Journal of Research and Development*. 2002. Vol. 46. № 2/3. P. 265–286. DOI: 10.1147/rd.462.0265.
10. Hu C., Tam S. C., Hsu F.-C., Ko P.-K., Chan T.-Y., Terrill K. W. Hot-electron-induced MOSFET degradation – model, monitor, and improvement // *IEEE Journal of Solid-State Circuits*. 1985. Vol. 20. № 1. P. 295–305. DOI: 10.1109/JSSC.1985.1052306.
11. Новоселов А. С., Масальский Н. В. Влияние деградации горячих носителей на характеристики высоковольтного КНИ транзистора с большой областью дрейфа // *Микроэлектроника*. 2023. Т. 52. № 5. С. 423–430. DOI: 10.31857/S0544126923700497.
12. Харитонов И. А. SPICE-модели МОПТ, учитывающие эффекты старения // *Наноиндустрия*. 2020. Т. 13. № 3s. С. 300–307. DOI: 10.22184/1993-8578.2020.13.3s.300.307.
13. Харитонов И. А., Белопашенцев А. С. Влияние эффектов старения на электрические характеристики интегральных КМОП ОУ при уменьшении минимальных размеров транзисторов // *Наноиндустрия*. 2022. Т. 15. № S8-I (113). С. 195–199. DOI: 10.22184/1993-8578.2022.15.8s.195.199.
14. Wong H., Kakushima K. On the vertically stacked gate-all-around nanosheet and nanowire transistor scaling beyond the 5 nm technology node // *Nanomaterials*. 2022. Vol. 12. № 10. Article 1739. DOI: 10.3390/nano12101739.
15. Wang M. A Review of Reliability in Gate-All-Around Nanosheet Devices // *Micromachines*. 2024. Vol. 15. № 2. Article 269. DOI: 10.3390/mi15020269.
16. Khdr H., Amrouch H., Henkel J. Dynamic Guardband Selection: Thermal-Aware Optimization for Unreliable Multi-Core Systems // *IEEE Transactions on Computers*. 2019. Vol. 68. № 1. P. 53–66. DOI: 10.1109/TC.2018.2848276.

17. Huang K., Anik M. T. H., Zhang X., Karimi N. Real-Time IC Aging Prediction via On-Chip Sensors // 2021 IEEE Computer Society Annual Symposium on VLSI. 2021. P. 13–18. DOI: 10.1109/ISVLSI51109.2021.00014.
18. Haghbayan M. H., Miele A., Mutlu O., Plosila J. Run-Time Resource Management in CMPs Handling Multiple Aging Mechanisms // IEEE Transactions on Computers. 2023. Vol. 72. № 10. P. 2872–2887. DOI: 10.1109/TC.2023.3272800.
19. Sun Z., Kim T., Chow M., Peng S., Zhou H., Kim H., Wong D., Tan S. X.-D. Long-Term Reliability Management for Multitasking GPGPUs // 2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design. 2019. P. 213–216. DOI: 10.1109/SMACD.2019.8795268.
20. Leng J., Buyuktosunoglu A., Bertran R., Bose P., Zu Y., Reddi V. J. Predictive Guardbanding: Program-Driven Timing Margin Reduction for GPUs // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2021. Vol. 40. № 1. P. 171–184. DOI: 10.1109/TCAD.2020.2992684.
21. Zhang J., Sadiqbatcha S., Tan S. X.-D. Hot-Trim: Thermal and Reliability Management for Commercial Multicore Processors Considering Workload Dependent Hot Spots // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2023. Vol. 42. № 7. P. 2290–2302. DOI: 10.1109/TCAD.2022.3216552.
22. Ghasemi S. M., Krautter J., Gheshlaghi T., Mescikov S., Gnad D. R. E., Tahoori M. B. Degradation Monitoring Through Software-Controlled On-Chip Sensors for RISC-V // 2024 IEEE European Test Symposium. 2024. P. 1–6. DOI: 10.1109/ETS61313.2024.10567607.
23. Balef H. A., Goossens K., Pineda de Gyvez J. Chip Health Tracking Using Dynamic In-Situ Delay Monitoring // 2019 Design, Automation & Test in Europe Conference & Exhibition. 2019. P. 304–307. DOI: 10.23919/DATE.2019.8715014.
24. Ofori-Attah E., Agyeman M. O. An Ageing-Aware and Temperature Mapping Algorithm for Multilevel Cache Nodes // IEEE Access. 2023. Vol. 11. P. 19162–19172. DOI: 10.1109/ACCESS.2022.3174084.
25. Hossein Khani F., Akbari O., Shafique M. A Two-Level Thermal Cycling-Aware Task Mapping Technique for Reliability Management in Manycore Systems // IEEE Access. 2024. Vol. 12. P. 113406–113421. DOI: 10.1109/ACCESS.2024.3443539.
26. Ramadan F., Ganaeim M., Ella M., Gabbay F. The Impact of Asymmetric Transistor Aging on Clock Tree Design Considerations // IEEE Access. 2024. Vol. 12. P. 177781–177794. DOI: 10.1109/ACCESS.2024.3506059.
27. Amrouch H., Ehsani S. B., Gerstlauer A., Henkel J. On the Efficiency of Voltage Overscaling under Temperature and Aging Effects // IEEE Transactions on Computers. 2019. Vol. 68. № 11. P. 1647–1662. DOI: 10.1109/TC.2019.2916869.
28. Bian S., Shintani M., Morita S., Awano H., Hiromoto M., Sato T. Workload-Aware Worst Path Analysis of Processor-Scale NBTI Degradation // Proceedings of the 26th Edition of the Great Lakes Symposium on VLSI. 2016. P. 203–208. DOI: 10.1145/2902961.2903013.
29. Zhang Z., Guo Z., Lin Y., Li M., Wang R., Huang R. AVATAR: An Aging- and Variation-Aware Dynamic Timing Analyzer for Error-Efficient Computing // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2023. Vol. 42. No. 11. P. 4139–4151. DOI: 10.1109/TCAD.2023.3255167.
30. Raji M., Mahmoudi R., Ghavami B., Keshavarzi S. Lifetime Reliability Improvement of Nano-Scale Digital Circuits Using Dual Threshold Voltage Assignment // IEEE Access. 2021. Vol. 9. P. 114120–114134. DOI: 10.1109/ACCESS.2021.3103200.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.



ОБЗОР ИССЛЕДОВАНИЙ ПО ПРОБЛЕМЕ ОЦЕНКИ ПРОИЗВОДСТВЕННО-ТЕХНОЛОГИЧЕСКИХ РИСКОВ НЕФТЯНОГО МЕСТОРОЖДЕНИЯ

Белоусов А. С., Астапов В. Н. ORCID ID 0000-0002-7979-5322, Мельникова Д. А.

*Федеральное государственное бюджетное образовательное учреждение высшего образования
«Самарский государственный технический университет», Самара, Российская Федерация,
e-mail: anbelousov2006@gmail.com*

В работе выполнен анализ научных публикаций и нормативной документации по теме оценки производственно-технологических рисков нефтяных месторождений на этапе их эксплуатации. Обобщены концептуальные подходы к риск-менеджменту в нефтегазодобыче: теория надёжности промышленного оборудования, принципы промышленной безопасности, вероятностный анализ риска, экономические модели ущерба и риск-ориентированное техническое обслуживание (RCM/RBI). Предложена классификация рисков сразу по нескольким основаниям: источнику угрозы (внешние/внутренние), стадии жизненного цикла актива, физической природе воздействия (механические, коррозионно-эрозийные, термодинамические, химико-технологические, электротехнические, программно-аппаратные), масштабу последствий и степени управляемости; производственно-технологические риски выделены как центральный элемент внутреннего риск-профиля нефтегазового проекта. Дана характеристика ключевых аналитических методов: качественных (HAZID, HAZOP, FMEA/FMECA, матрицы риска, Bow-Tie, FTA, ETA) и количественных (деревья отказов и событий, марковские модели, метод Монте-Карло, сценарный анализ), с примерами их применения (HAZOP, FTA/ETA, RBI) на реальных нефтегазовых объектах. Отдельно рассмотрены цифровые двойники скважин и установок подготовки нефти как современный инструмент управления рисками, а также прикладное программное обеспечение для автоматизации риск-анализа – PHA-Pro, PHAWorks, HAZOP Manager, Petex GAP, DNV Phast/SAFETI. Сделан вывод о необходимости интегрированного, многоуровневого подхода к оценке рисков и обновления отраслевой нормативной базы с учётом требований Ростехнадзора и действующих ГОСТов.

Ключевые слова: нефтяное месторождение, промышленное оборудование, производственно-технологический риск, оценка рисков, промышленная безопасность, HAZOP, FTA, RBI, цифровой двойник, нефтегазовая отрасль

REVIEW OF RESEARCH ON THE PROBLEM OF ASSESSING PRODUCTION AND TECHNOLOGICAL RISKS OF AN OIL FIELD

Belousov A. S., Astapov V. N. ORCID ID 0000-0002-7979-5322, Melnikova D. A.

*Federal State Budgetary Educational Institution of Higher Education
“Samara State Technical University”, Samara, Russian Federation,
e-mail: anbelousov2006@gmail.com*

This review examines the current body of scientific literature and regulatory documentation on assessing production and technological risks at oil fields during operation. It brings together the conceptual foundations of risk management used by oil and gas companies – reliability theory for field equipment, industrial safety principles, probabilistic risk analysis, economic loss models, and risk-based maintenance and inspection (RCM/RBI). Risks are classified along several dimensions at once: by the nature of the hazard source (external or internal), by the stage of the asset life cycle, by the physical nature of the impact (mechanical, corrosion-erosion, thermohydraulic, chemical-technological, electrical, and hardware/software), and by the scale of consequences and degree of controllability, with production and technological risk singled out as a core element of an oil and gas project's internal risk profile. Both qualitative techniques (HAZID, HAZOP, FMEA/FMECA, risk matrices, Bow-Tie, FTA, ETA) and quantitative ones (fault and event trees, Markov models, Monte Carlo simulation, scenario analysis) are discussed, together with case examples of HAZOP, FTA/ETA, and RBI applied at real oilfield facilities. The paper also looks at digital twins of wells and oil-treatment units as a modern risk-management tool, and surveys applied software used to automate risk analysis – PHA-Pro, PHAWorks, HAZOP Manager, Petex GAP, and DNV Phast/SAFETI. The review concludes that risk assessment in this sector needs an integrated, multilevel approach, alongside an update of industry regulations to reflect Rostekhnadzor requirements and current GOST standards.

Keywords: oil field, oilfield equipment, production and technological risk, risk assessment, industrial safety, HAZOP, FTA, RBI, digital twin, oil and gas industry

Введение

Разработка и эксплуатация нефтяных месторождений сопряжены с множеством технологических угроз: от отказов оборудования до ошибок персонала, каждая из кото-

рых потенциально способна затронуть людей, промышленные объекты, окружающую среду и экономику проекта. Рост инвестиций в отрасль, усложнение технологических схем и ужесточение регуляторных требова-

ний делают системное управление рисками всё более насущной практической задачей. Далее под производственно-технологическим риском понимается вероятность нежелательного события – отказа оборудования, отклонения от заданных параметров процесса или ошибки персонала, – рассматриваемая совместно с тяжестью его возможных последствий [1; 2].

В России нормативную базу для оценки риска на опасных производственных объектах (ОПО) задают Федеральный закон № 116-ФЗ «О промышленной безопасности опасных производственных объектов» [1] и методические указания Ростехнадзора по анализу риска – РД 03-418-01 [2]. Формально этот документ утратил статус нормативного, но на практике продолжает применяться как рекомендательное руководство и сохраняет свою методическую ценность. К этому добавляется набор действующих ГОСТов: Р ИСО 17776-2012 «Нефтяная и газовая промышленность. Морские добычные установки. Способы и методы идентификации опасностей и оценки риска. Основные положения» [3], Р ИСО 31000-2019 «Менеджмент риска. Принципы и руководство» [4], Р МЭК 61511-1-2018 «Безопасность функциональная. Системы безопасности приборные для промышленных процессов» [5] и Р МЭК 60812-2022 «Анализ видов и последствий отказов (FMEA и FMECA)» [6], а также прочие документы Ростехнадзора, регламентирующие промышленную безопасность [1; 2].

Цель работы – проанализировать современные исследования и нормативные документы, посвящённые методам оценки производственно-технологических рисков при разработке и эксплуатации нефтяных месторождений. Для этого решается несколько задач: систематизировать концептуальные основы риск-менеджмента в нефтегазовой отрасли, рассмотреть существующие подходы к классификации производственно-технологических рисков, охарактеризовать качественные и количественные методы их оценки, а также рассмотреть предиктивную аналитику, цифровые двойники и программные средства автоматизации риск-анализа применительно к объектам нефтедобычи.

Материалы и методы исследования

Работа построена как аналитический обзор литературы. Источники отбирались по базам eLIBRARY.RU, Scopus, Web of Science и Google Scholar по запросам на русском и английском языках, включавшим термины «нефтяное месторождение», «производственно-технологический риск», «промышленная безопасность», «RBI»,

«HAZOP», «FTA», «цифровой двойник», «предиктивное обслуживание», а также их англоязычные эквиваленты: oilfield risk assessment, process safety, asset integrity, digital twin oil and gas. Хронологические рамки отбора – 2003–2026 годы. В обзор включались публикации, соответствующие тематике оценки рисков на объектах нефтегазодобычи и прошедшие рецензирование (для статей) либо имеющие официальный статус (для нормативных документов); всего был отобран 21 источник.

Методически исследование опирается на системный анализ отобранных источников, вероятностно-статистический аппарат для описания неопределённости, сопоставление качественных и количественных методов оценки риска, а также на типологизацию рисков нефтяного месторождения сразу по нескольким признакам: источнику угрозы, стадии жизненного цикла, физической природе воздействия, масштабу последствий и степени управляемости. Отдельно обобщены сведения о программных средствах автоматизации риск-анализа. Риск здесь понимается традиционно – как произведение вероятности нежелательного события на величину его последствий для людей, окружающей среды, оборудования и финансовых показателей проекта [4; 7].

Результаты исследования и их обсуждение

1. Теоретические основы и классификация рисков нефтяного месторождения

Сегодня риск-менеджмент – обязательный элемент технического и производственного управления месторождением на всех этапах его жизни, от геологоразведки до ликвидации и рекультивации. В основе управления рисками при разработке залежей лежат системно-процессный анализ, вероятностно-статистические методы описания неопределённости и принципы промышленной и экологической безопасности [7]. На практике риск инженеры выражают двояко: либо через математическое ожидание потерь, либо через безразмерные матрицы «вероятность – последствие» с заданными порогами приемлемости и критичности [8].

Риски в нефтегазовой сфере принято разделять по нескольким основаниям. По масштабу охвата – на индивидуальные (затрагивающие конкретного работника или единицу оборудования) и социальные (затрагивающие группу людей, населённый пункт, экосистему). По предметной области – на производственно-технологические, геологические, экономические, экологические и организационно-управленческие [9].

Применительно к месторождению производственно-технологический риск – это вероятность отказа оборудования, отклонения параметров процесса или управленческой ошибки, следствием которых являются остановка добычи, снижение качества продукции, авария или инцидент, связанный с безопасностью. Исследования в области asset integrity management показывают, что именно на эту группу рисков приходится значимая доля нештатных ситуаций при эксплуатации скважин и установок подготовки нефти [10].

Оценка производственно-технологических рисков опирается на пять взаимодополняющих теоретических блоков.

Первый блок – теория надёжности технических систем: она описывает, как со временем деградирует и отказывает промышленное оборудование, и оперирует такими показателями, как интенсивность отказов, наработка на отказ, ремонтпригодность и коэффициент готовности; на уровне объекта в целом применяются модели последовательных, параллельных и смешанных структур, а также блок-схемы надёжности [11]. Второй блок – инженерия промышленной безопасности, оперирующая понятиями опасности, угрозы, уязвимости и защитного барьера; здесь центральную роль играют процедуры HAZID, HAZOP и FMEA/FMECA, закреплённые в ГОСТ Р МЭК 60812-2022 [6]. Третий блок – вероятностный риск-анализ: деревья отказов (FTA) и событий (ETA), марковские модели состояний, имитационное моделирование по методу Монте-Карло, сценарный анализ и анализ чувствительности [11]. Четвёртый блок – экономика риска: она связывает технологические сбои с потерей добычи, недополученной выручкой и штрафами, а также используется для оптимизации затрат по принципу ALARP. Пятый блок – управление активами и риск-ориентированное обслуживание (RCM/RBI), где приоритетность, объём и периодичность инспекций определяются риск-профилем конкретного узла [10].

Отдельного внимания заслуживает разграничение типов неопределённости: в прикладных задачах выделяют алеаторическую неопределённость, связанную со случайной природой явлений, и эпистемическую – связанную с нехваткой знаний об объекте [11]. Кроме того, при оценке риска на промысле важно учитывать отказы по общей причине: нарушение электроснабжения способно одновременно вывести из строя сразу несколько технологических узлов, а некачественный ингибитор коррозии – ускорить деградацию на разных участках трубопровода. Отсюда вытекает необходимость моделирования каскадных эффектов в структурных моделях надёжности [11; 12].

Классификация рисков нефтяного месторождения нужна для того, чтобы охватить всё многообразие угроз и сделать их сопоставимыми между собой и приоритизированными [3; 9]. Прежде всего риски делят на внешние и внутренние. К внешним относят макроэкономические и рыночные факторы, природно-климатические воздействия, форс-мажор – эта группа в настоящем обзоре подробно не рассматривается. Внутренние риски (геологические, производственно-технологические, организационно-управленческие, экологические), напротив, находятся в зоне ответственности оператора месторождения и управляются инженерными решениями, регламентами и культурой безопасности.

Если смотреть на жизненный цикл месторождения, в нём выделяют стадии геологического изучения, оценки и подготовки запасов, проектирования обустройства, строительства инфраструктуры, эксплуатации и поддержания добычи, модернизации и интенсификации, а также вывода из эксплуатации и рекультивации [7]. По физической природе воздействия производственно-технологические риски делят на механические (износ и усталостное разрушение трубопроводов, насосного оборудования), коррозионно-эрозионные (разрушение стенок аппаратов и скважинных труб под действием пластовых флюидов), термогидравлические (гидратообразование, парафиновые отложения, гидроудары), химико-технологические (загазованность, сбои в режимах подготовки нефти), а также электротехнические и программно-аппаратные [6; 9]. По масштабу последствий их разделяют на локальные (отдельная скважина, узел), объектовые (установка подготовки нефти, куст скважин) и системные, когда каскадный отказ распространяется через общие ресурсы вплоть до уровня месторождения. Наконец, по степени управляемости различают контролируемые риски, снижаемые инженерными и цифровыми методами, и практически неуправляемые, для которых остаётся полагаться главным образом на устойчивое проектирование и страхование [4].

2. Методы оценки и анализа рисков нефтяных проектов

В научно-технической литературе оценку рисков нефтяных проектов принято описывать как многоуровневую процедуру, которая сочетает качественные и количественные методы, вероятностно-статистическое моделирование, экономическую оценку последствий и инструменты предиктивной аналитики [3; 7].

Основа любой риск-программы – качественные методы: именно они формируют каталог угроз и описывают причинно-следственные цепочки [8; 9]. Среди них: идентификация опасностей (HAZID), анализ опасностей и работоспособности (HAZOP), анализ видов, последствий и критичности отказов (FMEA/FMECA) [6], матрицы «вероятность – следствие», метод «галстук-бабочка» (Bow-Tie), деревья отказов (FTA) и событий (ETA), а также структурированные экспертные процедуры [13].

Применение этих методов на объектах нефтегазовой отрасли достаточно подробно освещено в литературе. Так, HAZOP обычно применяют при анализе технологических схем установок подготовки и перекачки нефти: рабочая группа из технолога, механика, специалиста КИПиА и эксперта по безопасности совместно выявляет нетипичные отклонения давления, температуры и расхода, которые легко упустить при стандартной инженерной проверке [9]. Метод FTA широко используется при оценке надёжности противофонтанного оборудования (БОП): дерево отказов, построенное до базовых событий с известными интенсивностями отказов компонентов, позволяет количественно оценить вероятность потери контроля над скважиной и выделить наиболее значимые причины [11]. Подход RBI (Risk-Based Inspection), в свою очередь, показал себя эффективным при управлении целостностью трубопроводов и резервуаров: по данным ряда исследований, переход от фиксированных регламентных интервалов к графику, основанному на риск-профиле, способен сократить число внеплановых остановок и одновременно оптимизировать расходы на инспекции [10].

В основе количественного анализа риска лежат формализованные модели причинно-следственных связей [11; 14]: зная интенсивности отказов элементарных событий, можно рассчитать вероятность или частоту наступления целевого события за заданный период, проверить чувствительность модели и найти «минимальные сечения» – наборы причин, вносящих основной вклад в результат. Для оценки инвестиционных и операционных рисков здесь чаще всего применяют сценарный анализ и имитационное моделирование методом Монте-Карло [10; 11].

Человеческий фактор учитывают через деревья ошибок оператора: базовым действиям присваивают вероятности, скорректированные на уровень стресса, опыт персонала, доступность рабочих инструкций и удобство операторских интерфейсов [6]. Кроме того, в количественном анализе риска применяется «барьерный» подход, за-

креплённый в ГОСТ Р МЭК 61511-1-2018: функции безопасности (SIF) должны обеспечивать требуемый уровень полноты безопасности (SIL) с учётом вероятности отказа по требованию и архитектуры конкретной системы [5].

Отдельное и быстро развивающееся направление – применение анализа данных и машинного обучения в управлении рисками промышленного оборудования [15]. Задачу предиктивного обслуживания здесь формулируют как прогнозирование остаточного ресурса (RUL) и классификацию технического состояния по данным сенсорных потоков. Важная оговорка: результативность ML-моделей сильно зависит от качества и полноты исторических данных об отказах, а при небольших или несбалансированных выборках их дополнительно нужно проверять на независимых данных [15].

Цифровые двойники скважин и установок подготовки нефти совмещают первопринципные гидродинамические и технологические модели с данными оперативного мониторинга, что даёт возможность прогонять сценарии «что, если...» почти в реальном времени [16]. Отдельные исследования показывают, что связка цифрового двойника с системой управления техническим обслуживанием (CMMS) и базой данных отказов помогает раньше замечать деградацию оборудования, однако для масштабного промышленного внедрения таких решений нужна стандартизация обмена данными и соответствующая ИТ-инфраструктура [16; 17].

В литературе неоднократно отмечается, что качество дата-ориентированных подходов напрямую зависит от исходных данных, единообразия классификаторов отказов и открытости отчётности об инцидентах на предприятии. Поэтому для задач промышленной безопасности чаще рекомендуют «гибридное» моделирование, где физические модели задают допустимые границы решения, а машинное обучение уточняет детали внутри этих границ [15]. Экономически результаты риск-анализа интерпретируют через расчёт ожидаемых потерь и стоимость управляющих мер, соотносённую с принципом ALARP [11].

В целом сложившуюся практику оценки рисков нефтяных проектов можно описать как трёхуровневую интегрированную методологию. Нижний уровень – качественные методы HAZID/HAZOP/FMEA, формирующие полный реестр угроз и защитных барьеров. Средний уровень – FTA/ETA, Bow-Tie, марковские модели и блок-схемы надёжности, дающие количественное описание причинно-следственных связей. Верхний уровень – имитационные расчёты Монте-

Карло, сценарный анализ и анализ чувствительности, переводящие технические вероятности в распределения финансовых показателей. Вдобавок ко всем трём уровням постоянно работают RCM/RBI и цифровые двойники с элементами машинного обучения, которые корректируют оценки по мере поступления эксплуатационных данных [10; 16]. Такая структура позволяет увязывать инженерные решения с бизнес-целями и формировать портфель управляющих мер по принципу ALARP в соответствии с нормативными требованиями [1; 2; 5].

3. Программные средства для оценки рисков

Программное обеспечение для оценки рисков в нефтегазовой отрасли образует своего рода многоуровневую экосистему: она охватывает и инженерно-технические аспекты промышленной безопасности, надёжности и целостности активов, и финансово-экономическую неопределённость инвестиционных решений. Базовый уровень занимают PHA-инструменты, которые стандартизируют и документируют качественные идентификацию опасностей.

Такие пакеты, как PHA-Pro (разработчик – Sphera, ранее Dyadem International) [18], PHAWorks (разработчик – Primatech) [19] и HAZOP Manager (разработчик – Lihou Technical and Software Services) [20], дают шаблоны технологических узлов, библиотеки направляющих слов и типовых отклонений, автоматически формируют перечень причин, последствий и действующих барьеров, а также помогают отслеживать корректирующие мероприятия. Общие тенденции автоматизации подобных PHA-систем обобщены в отраслевых литературных обзорах [21]. Отдельно стоит отметить прослеживаемость данных в современных PHA-системах: одно и то же опасное отклонение последовательно фигурирует в базе знаний, технологической схеме, Bow-Tie-диаграмме и количественной модели LOPA, где для каждого сценария рассчитывается независимость барьеров и требуемый уровень полноты безопасности (SIL) [5].

Для моделирования последствий аварийных выбросов и взрывов на нефтегазовых объектах чаще всего используют комплекс DNV Phast/SAFETI (DNV GL): он рассчитывает зоны поражения при разных сценариях разгерметизации, моделирует рассеивание токсичных и горючих газов и формирует отчётность, соответствующую требованиям промышленно-безопасного законодательства [17]. Гидравлическое моделирование добычных систем и оценку режимных рисков скважин, в свою очередь, чаще выпол-

няют в Petex GAP (Petroleum Experts) – программе, которая помогает выявлять зоны возможного гидратообразования, отложений и нестабильного режима течения.

Внедрение программных средств риск-анализа требует продуманной архитектуры. Сначала нужно определить единый «эталонный источник» данных – общий репозиторий паспортов оборудования, журналов отказов, результатов неразрушающего контроля, телеметрии и иерархий активов. Затем – наладить двусторонний обмен данными между риск-платформой и системами CMMS, SCADA, LIMS, ERP. И наконец – собрать общий реестр сценариев и защитных барьеров, который поддерживал бы «живой» пересчёт риска по мере поступления данных о новых инцидентах или результатах инспекций [5].

Одно из перспективных направлений – объединение PHA, QRA, RBI и APM в единые платформы с непрерывным пересчётом риска и автоматической приоритизацией технического обслуживания. Ряд исследований указывает, что при наличии валидированных исторических данных об отказах такой подход способен сократить число незапланированных остановок и оптимизировать бюджет на инспекции; правда, для его реализации на практике нужны зрелая культура сбора данных и интеграция между системами [16; 17].

Заключение

Проведённый обзор показывает, что оценка производственно-технологических рисков нефтяного месторождения – вполне зрелая область, в которой накоплен широкий инструментарий: от качественной идентификации опасностей до количественных вероятностных моделей и предиктивной аналитики на данных. При этом ключевое условие результативной оценки – системность: угрозы нужно идентифицировать на всех стадиях жизненного цикла объекта, классифицировать по соответствующим признакам, а метод оценки подбирать исходя из природы конкретного риска и объёма доступных данных.

Среди внутренних рисков нефтегазового проекта производственно-технологические занимают центральное место: отказы оборудования и нарушения технологического режима способны привести к серьёзному экономическому ущербу, угрозе жизни персонала и вреду для окружающей среды. Рассмотренные практические кейсы применения HAZOP, FTA/ETA и RBI подтверждают, что своевременное выявление уязвимостей в промышленном оборудовании и трубопроводах помогает сократить чис-

ло аварийных остановок и снизить затраты на ликвидацию последствий инцидентов.

При наличии качественных валидированных данных цифровые двойники, ML-модели прогнозирования остаточного ресурса и интегрированные программные платформы способны заметно повысить оперативность и обоснованность управленческих решений в области промышленной безопасности. При этом часть методологических вопросов пока остаётся открытой: нужны дальнейшая адаптация моделей к специфике российских нефтегазовых объектов, обновление нормативной базы взамен устаревшего РД 03-418-01, а также гармонизация отечественных стандартов с международными требованиями в области asset integrity management. Работа в этом направлении напрямую связана с повышением надёжности и устойчивости нефтегазовых проектов в целом.

Список литературы

1. Федеральный закон от 21.07.1997 № 116-ФЗ «О промышленной безопасности опасных производственных объектов» // Официальный интернет-портал правовой информации. Актуальная ред. от 08.08.2024. URL: <https://docs.cntd.ru/document/9046058> (дата обращения: 15.07.2026).
2. РД 03-418-01. Методические указания по проведению анализа риска опасных производственных объектов. М.: Госгортехнадзор России, 2001. 26 с. (документ утратил нормативный статус; используется как методическое пособие). URL: <https://docs.cntd.ru/document/1200012878> (дата обращения: 15.07.2026).
3. ГОСТ Р 72192-2025 (ИСО 17776:2016). Нефтяная и газовая промышленность. Сооружения нефтепромысловые морские. Управление опасностями крупных аварий при проектировании. М.: ФГБУ «Институт стандартизации», 2025. Введ. 2026-03-31. 86 с. URL: <https://protect.gost.ru/gost/details/0aff0d561-0065-47fe-b4d7-1db5acf71077> (дата обращения: 22.08.2026).
4. ГОСТ Р ИСО 31000-2019. Менеджмент риска. Принципы и руководство. М.: Стандартинформ, 2019. 16 с. URL: <https://docs.cntd.ru/document/1200170125> (дата обращения: 15.07.2026).
5. ГОСТ Р МЭК 61511-1-2018. Безопасность функциональная. Системы безопасности приборные для промышленных процессов. Часть 1. Термины, определения и технические требования. М.: Стандартинформ, 2018. Введ. 2019-07-01. 126 с. URL: <http://docs.cntd.ru/document/1200160087> (дата обращения: 22.08.2026).
6. ГОСТ Р МЭК 60812-2022. Менеджмент риска. Анализ видов и последствий отказов (FMEA и FMCA). М.: Стандартинформ, 2022. Введ. 2022-11-01. 72 с. URL: <https://docs.cntd.ru/document/1200193512> (дата обращения: 15.07.2026).
7. Ахметзянова И. С., Глазкова И. Н. Методы управления рисками инвестиционных проектов разработки нефтегазовых месторождений // Булатовские чтения: сб. ст. / АГНИ. Альметьевск. 2020. С. 193–198. URL: <https://elibrary.ru/item.asp?id=44068493> (дата обращения: 15.07.2026).
8. Oudina Z., Dib A., Yakoubi M. A., Derdour M. Comprehensive risk classification and mitigation in the petroleum cyber-physical systems of the oil and gas industry // International Journal of Safety and Security Engineering. 2024. Vol. 14. № 1. P. 99–113. DOI: 10.18280/ijss.140110.
9. Khadem M. M. R. K., Piya S., Shamsuzzoha A. Quantitative risk management in gas injection project: a case study from Oman oil and gas industry // Journal of Industrial Engineering International. 2018. Vol. 14. P. 637–654. DOI: 10.1007/s40092-017-0237-3.
10. API RP 581. Risk-Based Inspection Methodology. 3rd ed. Washington, DC: American Petroleum Institute, 2016. 1032 p. ISBN: 978-0-7680-8296-7.
11. Aven T. The Science of Risk Analysis: Foundation and Practice. Abingdon; New York: Routledge, 2020. 330 p. ISBN: 978-0-367-13922-3. DOI: 10.4324/9780429029189.
12. Rausand M., Barros A., Hoyland A. System Reliability Theory: Models, Statistical Methods, and Applications. 3rd ed. Hoboken, NJ: John Wiley & Sons, 2021. 864 p. ISBN: 978-1-119-37352-0. DOI: 10.1002/9781119373940.
13. Ab Rahim M. S., Reniers G., Yang M., Bajpai S. Risk assessment methods for process safety, process security and resilience in the chemical process industry: A thorough literature review // Journal of Loss Prevention in the Process Industries. 2024. Vol. 88. Art. 105274. DOI: 10.1016/j.jlp.2024.105274.
14. Mandal C. S., Agarwal M. A review on Quantitative Risk Assessments for Oil and Gas installations and changes in Risk Evaluation Techniques // Materials Today: Proceedings. 2024. Vol. 99. P. 145–153. DOI: 10.1016/j.matpr.2023.07.059.
15. Carvalho T. P., Soares F. A. A. M. N., Vita R., Francisco R. da P., Basto J. P., Alcalá S. G. S. A systematic literature review of machine learning methods applied to predictive maintenance // Computers & Industrial Engineering. 2019. Vol. 137. Art. 106024. DOI: 10.1016/j.cie.2019.106024.
16. Wanasinghe T. R., Wroblewski L., Petersen B. K., Gosine R. G., James L. A., De Silva O., Mann G. K. I., Warrian P. J. Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges // IEEE Access. 2020. Vol. 8. P. 104175–104197. DOI: 10.1109/ACCESS.2020.2998723.
17. DNV. Phast & SAFETI: Risk Modelling Software. Technical Reference Manual / DNV GL AS. Rev. 8.4. Høvik: DNV, 2022. URL: <https://www.dnv.com/services/consequence-and-risk-modelling-phast-and-safeti-1175> (дата обращения: 15.07.2026).
18. Sphera. PHA-Pro: Process Hazard Analysis (PHA) & HAZOP Software [Электронный ресурс]. URL: <https://sphera.com/solutions/process-safety-management/advanced-risk-assessment-software-and-services/process-hazard-analysis-pha-pro-hazop/> (дата обращения: 22.08.2026).
19. Primatech Inc. PHAWorks – HAZOP Software [Электронный ресурс]. URL: <https://www.primatech.com/software/phaworks> (дата обращения: 22.08.2026).
20. Lihou Technical and Software Services. HAZOP Manager V7.0 – Software for Hazop Studies and other PHA Reviews [Электронный ресурс]. URL: <https://www.lihoutech.com/> (дата обращения: 22.08.2026).
21. Single J. I., Schmidt J., Denecke J. State of research on the automation of HAZOP studies // Journal of Loss Prevention in the Process Industries. 2019. Vol. 62. Art. 103952. DOI: 10.1016/j.jlp.2019.103952.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.



ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА SIMINTECH ДЛЯ ОПТИМИЗАЦИИ ПЕРЕРАСПРЕДЕЛЕНИЯ ЗАГРУЗКИ ГАЗОТУРБИННЫХ ДВИГАТЕЛЕЙ

Ильичев В. Ю. ORCID ID 0000-0003-1017-5544

*Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)», Москва, Российская Федерация,
e-mail: patrol8@yandex.ru*

В статье рассмотрено практическое применение генетического алгоритма в отечественной среде имитационного моделирования для оптимального перераспределения электрической нагрузки между параллельно работающими газотурбинными двигателями. Актуальность работы обусловлена устойчивым ростом распределённой генерации, необходимостью снижения удельного расхода топлива и курсом на импортозамещение программного обеспечения. Объектом исследования выбрана реальная электростанция с тремя агрегатами газотурбинных установок мощностью 16 мегаватт, имеющими разную наработку и расходные характеристики. Построена верифицированная по архивам автоматизированной системы управления имитационная модель группы двигателей. Для оптимального перераспределения нагрузки разработан генетический алгоритм на языке программирования Python с использованием библиотеки эволюционных алгоритмов, интегрированный с моделью через обмен данными и формирующий уставки загрузки. Показано, что оптимальное распределение нагрузки даёт экономию природного газа на 3,7% при суммарном потреблении 30 мегаватт, а при 42 мегаватт – на 2,1% по сравнению с традиционным равномерным распределением. Генетический алгоритм демонстрирует уверенную сходимость за 40-60 поколений и высокую работоспособность в динамическом режиме. Интегральная оценка годовой экономии топлива составляет 2,5-3,5%, что соответствует нескольким миллионам рублей для одной станции. Полученные результаты подтверждают экономическую целесообразность внедрения эволюционных методов оптимизации в контуры управления реальными газотурбинными электростанциями.

Ключевые слова: генетический алгоритм, имитационное моделирование, газотурбинный двигатель, оптимизация загрузки, распределение нагрузки, расход топлива, параллельная работа, SimInTech

APPLICATION OF THE SIMINTECH GENETIC ALGORITHM FOR OPTIMIZING LOAD REDISTRIBUTION AMONG GAS TURBINE ENGINES

Ilichev V. Y. ORCID ID 0000-0003-1017-5544

*Federal State Autonomous Educational Institution of Higher Education
“Bauman Moscow State Technical University (National Research University)”,
Moscow, Russian Federation, e-mail: patrol8@yandex.ru*

This article examines the practical application of a genetic algorithm in a domestic simulation environment for the optimal redistribution of electrical load among parallel-operating gas turbine engines. The relevance of this study is driven by the steady growth of distributed generation, the need to reduce specific fuel consumption, and the strategic shift towards software import substitution. The research focuses on an actual power plant equipped with three gas turbine units rated at sixteen megawatts that have different operating hours and distinct consumption characteristics. A simulation model of the engine group was developed and verified against historical data from the plant's automated process control system. A genetic algorithm was developed in Python using an evolutionary algorithms library and integrated with the model through data exchange to generate optimal load setpoints. The results show that optimal load distribution yields a three point seven percent savings in natural gas at a total demand of thirty megawatts, and a two point one percent savings at forty-two megawatts compared to the traditional uniform distribution. The genetic algorithm demonstrates robust convergence within forty to sixty generations and high operational efficiency in dynamic mode. During step changes in demand, the transient process maintains an exact power balance. The overall estimated annual fuel savings range from two point five to three point five percent, translating to several million rubles per power plant. These results convincingly confirm the economic feasibility of successfully implementing evolutionary optimization methods into the control loops of real gas turbine power plants using domestic software.

Keywords: genetic algorithm, simulation modeling, gas turbine engine, load optimization, load distribution, fuel consumption, parallel operation, SimInTech

Введение

Современные исследования в области моделирования технических систем всё в большей степени опираются на методы интеллектуального анализа данных [1] и ап-

парат дискретных динамических систем [2]. Эти подходы формируют основу для перехода от эмпирических диспетчерских регламентов к формализованным контурам управления энергетическими объектами.

Распределённая генерация на базе газотурбинных и парогазовых установок занимает всё более заметное место в электроэнергетике благодаря высокой манёвренности, компактности и возможности работы на природном газе [3]. Оптимальное распределение активных мощностей между агрегатами традиционно решается методами математического программирования [4], однако на практике эта задача часто решается равномерным распределением нагрузки либо по диспетчерским таблицам, не учитывающим текущее техническое состояние агрегатов.

Между тем агрегаты даже одной серии имеют различные расходные характеристики, которые с наработкой расходятся всё сильнее, и равномерное распределение приводит к систематическому перерасходу топлива. Эксплуатация параллельно работающих газотурбинных двигателей должна учитывать технические ограничения, закреплённые правилами эксплуатации электрических станций и сетей [5]. Обзор метаэвристических методов оптимизации показывает их высокую эффективность при решении подобных электроэнергетических задач [6].

Мультиагентные модели энергоснабжающих систем позволяют учесть сложные взаимодействия между компонентами [7], а современные интеллектуальные системы обеспечивают адаптацию к изменяющимся условиям эксплуатации [8]. Для комплексной оценки работы энергетических объектов необходимо также моделирование пространства выбросов в атмосфере [9].

Характеристики газотурбинных двигателей содержат изломы, зоны запрещённой работы и нелинейности, что делает целевую функцию негладкой и многоэкстремальной. Именно здесь проявляются преимущества генетических алгоритмов, которые работают с популяцией решений, учитывают произвольные ограничения через штрафные функции и применимы к смешанным переменным. Эффективность подобных подходов подтверждается методикой построения баланса в гидротепловых энергосистемах [10], комплексной оптимизацией режимов промышленных тепловых электростанций [11] и применением имитационных моделей систем автоматического управления [12].

Отдельную актуальность задаче придаёт курс на импортозамещение программного обеспечения. Отечественная среда SimInTech содержит библиотеки для моделирования энергетических объектов, а параметрический синтез регуляторов с использованием метаэвристических алгоритмов в этой среде демонстрирует их работоспособность [13]. Оптимизация электрических сетей генетическими алгоритмами с актив-

но-адаптивными элементами подтверждает применимость эволюционных методов к энергосистемам [14], а корректный учёт ограничений обеспечивает техническую реализуемость получаемых решений [15].

Целью настоящей работы является практическая апробация генетического алгоритма отечественной среды имитационного моделирования SimInTech для задачи оптимального перераспределения электрической нагрузки между параллельно работающими газотурбинными двигателями при изменении суммарного потребления электрической энергии и количественная оценка достигаемой экономии топлива. Актуальность исследования обусловлена устойчивым ростом доли распределённой генерации на базе газотурбинных и парогазовых установок, необходимостью снижения удельного расхода топлива в условиях переменных графиков нагрузки, а также курсом на импортонезависимость программных средств технологического моделирования и управления.

Научная новизна состоит в том, что впервые для кластера ГТЭ-16 предложена архитектура замкнутого контура «модель станции – генетический алгоритм – уставки АСУ ТП», реализованная полностью на отечественном программном обеспечении и сочетающая защитный слой, гарантирующий соблюдение технических ограничений, и экономическое вознаграждение. Защитный слой реализован в виде штрафных функций за нарушение баланса мощности и за выход за допустимые диапазоны работы агрегатов, а функция вознаграждения минимизирует суммарный расход топлива. В отличие от известных подходов предложенная архитектура не требует пересмотра методики при изменении состава агрегатов или их характеристик после ремонта: достаточно обновить коэффициенты аппроксимации в модели.

Практическая значимость работы определяется курсом на импортонезависимость: весь контур реализуется на отечественном программном обеспечении и может быть интегрирован с АСУ ТП через стандартные протоколы обмена. Разработанная методика легко масштабируется.

Для достижения поставленной цели решаются следующие задачи: построение и верификация имитационной модели группы ГТД по реальным расходным характеристикам; формализация критерия оптимальности с учётом экономических компонент; настройка параметров генетического алгоритма и оценка его сходимости; проведение серии вычислительных экспериментов в стационарных и переходных режимах; сравнение результатов с традиционным равномерным распределением и оценка годовой экономии.

Материалы и методы исследования

Объектом исследования выбрана электростанция машиностроительного предприятия, включающая три параллельно работающих агрегата ГТЭ-16 с различной наработкой: № 1 – новый (после капитального ремонта), № 2 – около 50 тыс. ч, № 3 – около 80 тыс. ч. Агрегаты ГТЭ-16 номинальной мощностью 16 МВт подключены к общим шинам 10 кВ и работают параллельно на покрытие нагрузки предприятия с выдачей излишков во внешнюю сеть.

С ростом наработки ухудшается КПД турбомашин и растёт удельный расход топлива, а расходная характеристика смещается вверх и становится «круче». По данным испытаний, приведённых к стандартным условиям, расход газа каждого агрегата аппроксимирован квадратичной зависимостью от электрической мощности:

$$B_i(P_i) = a_i + b_i P_i + c_i P_i^2, \text{ м}^3/\text{ч},$$

где $i=1, 2, 3$ – номер агрегата; P_i – электрическая мощность i -го агрегата, МВт;

a_i, b_i, c_i – коэффициенты аппроксимации, полученные методом наименьших квадратов.

Для исследуемых агрегатов получены следующие коэффициенты:

агрегат № 1 (новый):

$$a_1=1200, b_1=850, c_1=12,5;$$

агрегат № 2 (50 тыс. ч):

$$a_2=1350, b_2=920, c_2=14,2;$$

агрегат № 3 (80 тыс. ч):

$$a_3=1480, b_3=980, c_3=15,8.$$

Критерий оптимальности (минимум суммарного расхода газа) $\min F = \sum B_i(P_i)$ при ограничениях:

$$\text{баланс мощности } \sum P_i = B,$$

где 1) D – суммарный спрос, МВт;

2) диапазон работы: $4 \leq P_i \leq 15,2$ МВт (не выше 95% от номинала 16 МВт);

3) темп набора нагрузки:

$$|dP_i / dt| \leq 0,5 \text{ МВт/с}.$$

Задача решается с использованием генетического алгоритма, реализованного на языке Python с применением библиотеки DEAP. Интеграция с моделью станции, построенной в среде имитационного моделирования SimInTech, осуществляется через обмен данными в CSV-формате: на каждом поколении генетический алгоритм формирует уставки загрузки агрегатов, которые передаются в модель станции; после моделирования переходного процесса значения расхода газа возвращаются в алгоритм для оценки приспособленности особей.

Генетический алгоритм настроен со следующими параметрами: размер популяции $N=60$ особей; число поколений $G=100$; вероятность кроссовера $p_c = 0,8$; вероятность мутации $p_m = 0,05$; метод селекции: турнирный отбор (размер турнира = 3); тип кроссовера: арифметический; тип мутации: гауссова с $\sigma = 0,1$.

Штрафные функции за нарушение ограничений имеют вид:

$$\Phi(P) = F(P) + \lambda_1 \left(\sum_{i=1}^3 P_i - D \right)^2 + \lambda_2 \sum_{i=1}^3 \max(0, 4 - P_i)^2 + \lambda_3 \sum_{i=1}^3 \max(0, P_i - 15,2)^2,$$

где $\lambda_1=1000, \lambda_2=500, \lambda_3=500$ – коэффициенты штрафов.

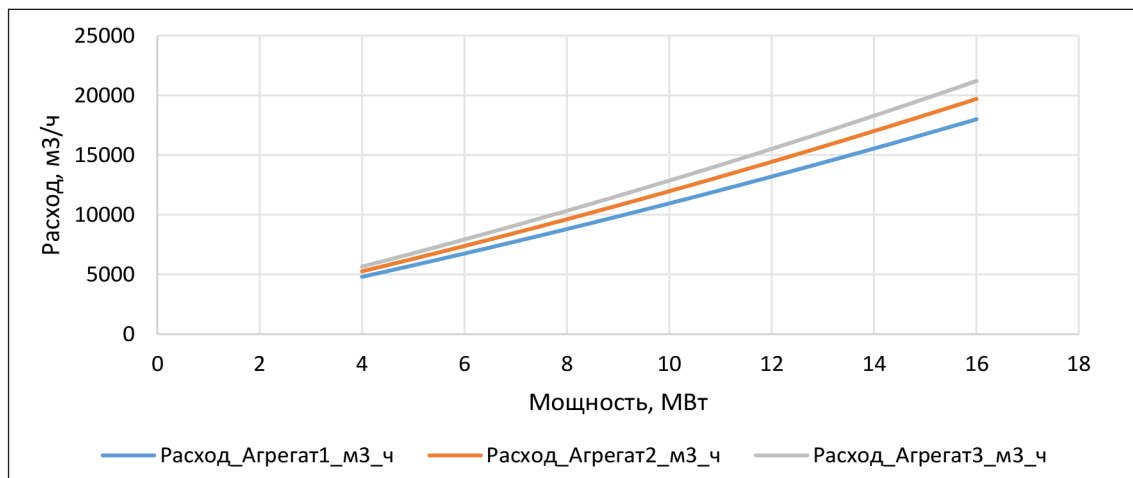


Рис. 1. Расходные характеристики параллельно работающих ГТЭ-16
Примечание: составлено авторами

Верификация модели проведена по архиву АСУ ТП за 6 месяцев эксплуатации. Средняя погрешность моделирования расхода газа составила 2,3%, что подтверждает адекватность модели для решения задачи оптимизации. Графическое сопоставление характеристик (рис. 1) показывает, что при мощности 16 МВт разница в расходе между новым и изношенным агрегатами достигает около 1 тыс. м³/ч, что и определяет экономический эффект оптимизации.

Результаты исследования и их обсуждение

На рис. 2 представлены кривые сходимости ГА для спроса 30 МВт: лучшая особь выходит на плато к 40–60-му поколению.

На рис. 3 показана эволюция популяции на ландшафте целевой функции: к 60-му поколению особи плотно группируются в окрестности глобального оптимума, что подтверждает механизм глобального поиска.

Сравнение равномерного и оптимального распределения показано на рис. 4.

Генетический алгоритм стремится загрузить наиболее экономичный агрегат № 1 до максимально допустимой нагрузки (95% номинала), агрегат № 2 – до зоны наилучшего относительного прироста, а наименее экономичный № 3 – к техническому минимуму при малом спросе. Это согласуется с принципом равенства относительных приростов, но получено автоматически, без аналитического вывода. Для спроса 30 МВт оптимальное распределение 15,2/9,5/5,3 МВт даёт экономию 3,7% (12,44 против 12,92 тыс. м³/ч), для 42 МВт распределение 15,2/15,2/11,6 МВт – 2,1% (16,93 против 17,29 тыс. м³/ч). При спросе 45,6 МВт и выше все агрегаты работают на максимально допустимой нагрузке 15,2 МВт, перераспределение невозможно и экономия равна нулю. Результаты динамического эксперимента (рис. 5): до 300 с станция работает при 30 МВт (распределение 15,2/9,5/5,3 МВт); после скачка до 42 МВт агрегаты выходят на новое распределение 15,2/15,2/11,6 МВт за 150–200 с с точностью баланса 0,2 МВт без перерегулирований.

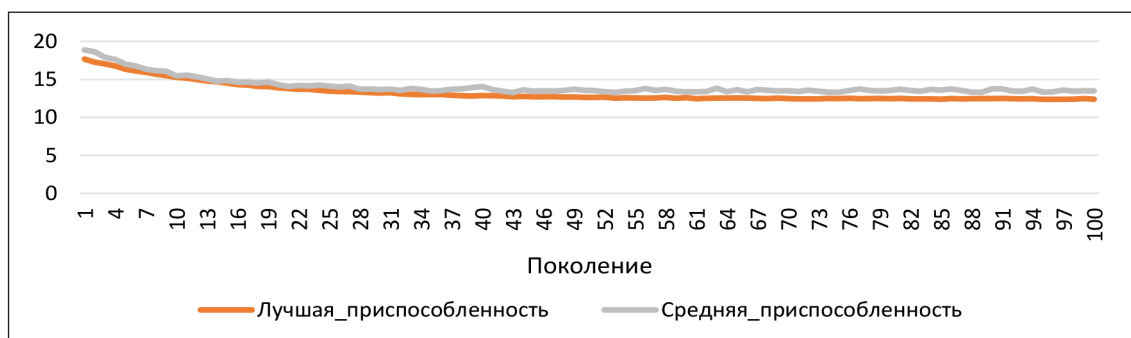


Рис. 2. Сходимость ГА: лучшая и средняя приспособленность (спрос 30 МВт)
Примечание: составлено авторами

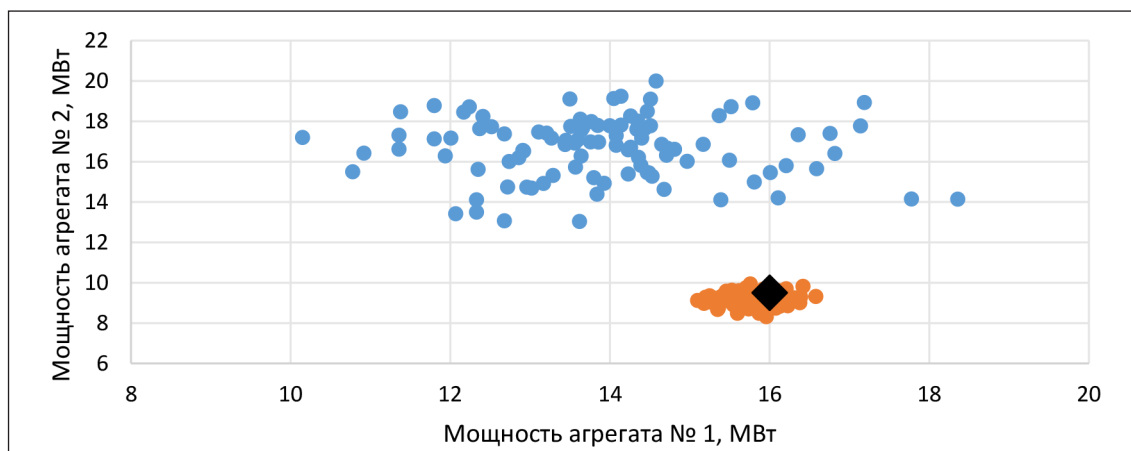


Рис. 3. Эволюция популяции ГА на ландшафте целевой функции ($D = 30$ МВт):
распределение особей на 10-м (синие точки) и 60-м (оранжевые точки) поколениях
Примечание: составлено авторами

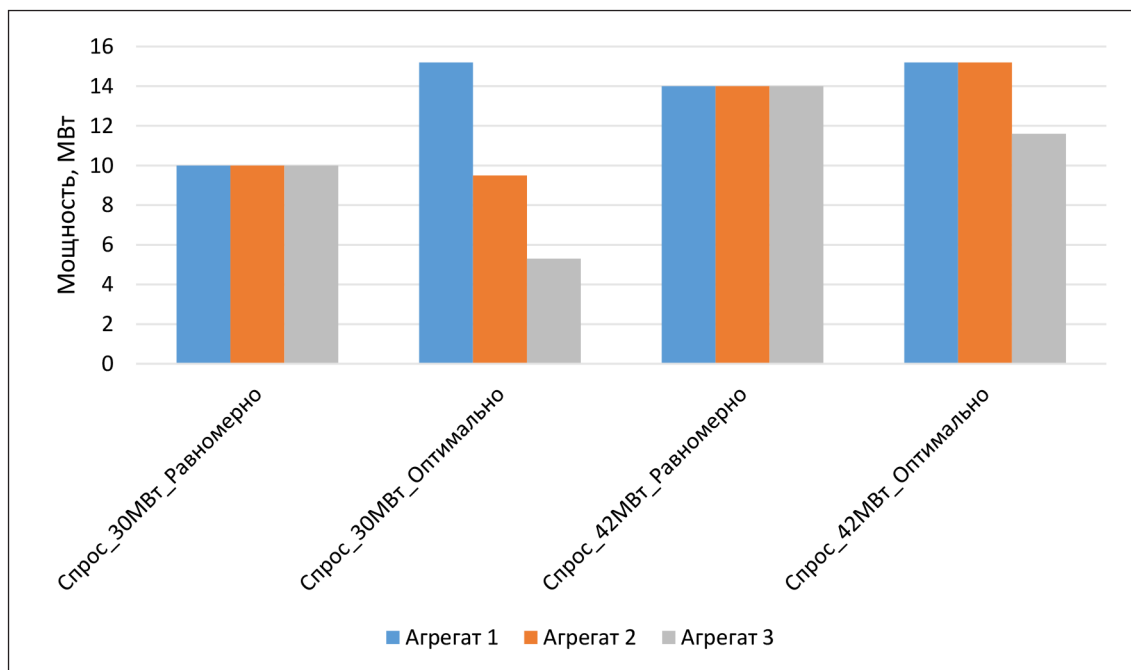


Рис. 4. Равномерное и оптимальное распределение при 30 МВт (10/10/10 против 15,2/9,5/5,3) и 42 МВт (14/14/14 против 15,2/15,2/11,6): экономия 3,7% и 2,1%
Примечание: составлено авторами

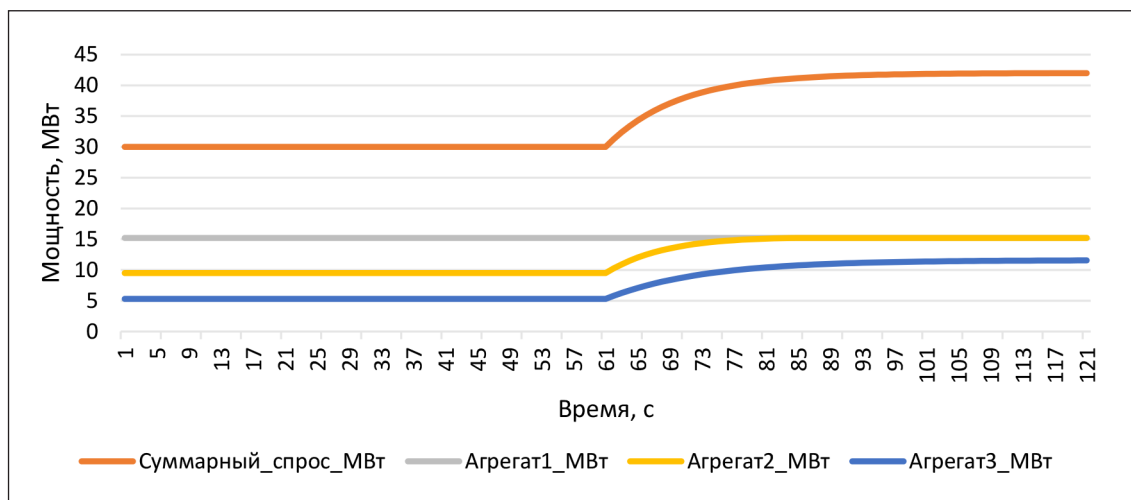


Рис. 5. Динамика перераспределения загрузки трёх ГТЭ-16 в SimInTech при скачке спроса
Примечание: составлено авторами

При скачке нагрузки агрегат № 1 остаётся на максимально допустимой нагрузке (15,2 МВт), а приращение оптимально распределяется между агрегатами № 2 и № 3, что подтверждает согласованную работу ГА-оптимизатора и динамических регуляторов; сходимость за 1–2 с (40–60 поколений) позволяет вести оперативное управление в реальном времени.

Экстраполяция на годовой график даёт экономию газа 2,5–3,5% (миллионы рублей

в год) при быстрой окупаемости: внедрение требует лишь доработки ПО верхнего уровня АСУ ТП. ГА обрабатывает негладкие характеристики и запрещённые зоны без ручного перебора, а после ремонта достаточно обновить коэффициенты модели; экономия 2–4% критична для энергетики, а недогрузка изношенного агрегата № 3 продлевает его ресурс. Подход согласуется с современными теориями оптимизации, а отечественная среда SimInTech отвечает требованиям

импортозамещения и обеспечивает независимость критической инфраструктуры.

Заключение

Таким образом, в работе апробирован генетический алгоритм, интегрированный с имитационной моделью станции в SimInTech, для оптимального перераспределения загрузки трёх ГТЭ-16. Экономия газа составляет 3,7% при спросе 30 МВт и 2,1% при 42 МВт; в годовом исчислении 2,5–3,5%, что соответствует нескольким миллионам рублей. ГА сходится за 40–60 поколений (1–2 с), перераспределение в динамике происходит безударно за 150–200 с.

Перспективы дальнейших научных исследований включают в себя уточнение существующей математической модели. В частности, планируется интеграция методов смешанного целочисленного нелинейного программирования для принятия оптимальных решений о пуске и останове агрегатов в зависимости от графика нагрузок. Также предусмотрено введение поправочных коэффициентов, учитывающих влияние температурно-климатических условий на характеристики оборудования. Завершающим этапом разработки станет опытно-промышленная апробация алгоритма на базе действующей газотурбинной электростанции.

Список литературы

1. Ильичев В. Ю., Морозова В. С. Оптимизация отбора признаков для качества прогнозирования в условиях высокой размерности данных // Научные технологии. 2021. Т. 22. № 4. С. 16–25. EDN: VNNTEX.
2. Ильичев В. Ю., Каширин Д. С. Исследование динамики поведения клеточных автоматов // E-Scio. 2023. № 1(76). С. 223–233. EDN: EZVNVE.
3. Варганова А. В. О методах оптимизации режимов работы электроэнергетических систем и сетей // Вестник Южно-Уральского государственного университета. Серия: Энергетика. 2017. Т. 17. № 3. С. 76–85. DOI: 10.14529/power170309. EDN: ZGQBYH.
4. Александров О. И., Лялько А. А. Решение задачи оптимального распределения активных мощностей в энергосистеме в среде MATLAB // Труды БГТУ. Серия 3: Физико-математические науки и информатика. 2019. № 1(218). С. 39–44. EDN: VWNELJ.
5. Правила технической эксплуатации электрических станций и сетей Российской Федерации: утв. приказом Минэнерго России от 04.10.2022 № 1070. М., 2023. 156 с.
6. Алехин Р. А., Кубарьков Ю. П., Закамов Д. В., Умаров Д. В. Обзор метаэвристических методов оптимизации, применяемых при решении электроэнергетических задач // Вестник Самарского государственного технического университета. Серия: Технические науки. 2019. № 3(63). С. 6–19. EDN: CWAUMA.
7. Стенников В. А., Барахтенко Е. А., Майоров Г. С. Разработка интегрированной мультиагентной модели энергонабжающей системы в программной среде AnyLogic // Вестник Иркутского государственного технического университета. 2020. Т. 24. № 5. С. 1080–1092. DOI: 10.21285/1814-3520-2020-5-1080-1092. EDN: IHYCBI.
8. Сиддигов И. Х., Порубай О. В., Лазарева М. В., Абдулхамидов А. А. Тенденции развития интеллектуальных систем при принятии управленческих решений в Узбекистане // Universum: технические науки. 2020. № 2-1(71). С. 10–13. EDN: CMMGOP.
9. Антонова А. М., Воробьев А. В., Воробьев В. А., Дутова Е. М., Покровский В. Д. Моделирование распространения в атмосфере загрязняющих веществ выбросов электростанций на базе программного комплекса «SKAT» // Известия Томского политехнического университета. 2019. Т. 330. № 6. С. 174–186. DOI: 10.18799/24131830/2019/6/2138. EDN: MZXAJR.
10. Секретарев Ю. А., Мятёж Т. В. Методика построения баланса в гидротепловой энергосистеме с учетом режимных особенностей ГЭС на основе определения стоимости гидроресурса // Журнал Сибирского федерального университета. Серия: Техника и технологии. 2021. Т. 14. № 6. С. 641–655. DOI: 10.17516/1999-494X-0340. EDN: YYMERL.
11. Варганова А. В., Джагаров Н. Ф. Комплексная оптимизация режимов работы промышленных тепловых электростанций // Электротехнические системы и комплексы. 2020. № 4(49). С. 11–16. DOI: 10.18503/2311-8318-2020-4(49)-11-16. EDN: MNSWRM.
12. Ковалёв Д. А., Шаряков В. А., Шарякова О. Л., Рубцова М. А. Разработка имитационной модели системы автоматического управления уровнем зелёного шёлока в баке-растворителе плава содорегенерационного котлоагрегата // Известия Санкт-Петербургского государственного технологического института (технического университета). 2023. № 67(93). С. 74–78. EDN: EYNWWD.
13. Михайленко Л. А., Еремин Е. Л. Параметрический синтез регулятора метаэвристическим алгоритмом в среде SimInTech // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2020. № 3. С. 45–56. DOI: 10.24143/2073-5529-2020-3-45-56. EDN: RKUNTE.
14. Кубарьков Ю. П., Макаров Я. В., Голубева К. А. Оптимизация электрической сети при помощи генетического алгоритма с внедрением активно-адаптивных элементов // Вестник Иркутского государственного технического университета. 2020. Т. 24. № 1. С. 112–122. DOI: 10.21285/1814-3520-2020-1-112-122. EDN: WOOVEN.
15. Гайбиров Т. Ш., Насиров Т. Х. Учет ограничений при оптимизации режимов энергосистем генетическими алгоритмами // Энергетика. Известия высших учебных заведений и энергетических объединений СНГ. 2020. Т. 63. № 4. С. 315–328. DOI: 10.21122/1029-7448-2020-63-4-315-328. EDN: MZXABC.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.



ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПОНЕНТА УПРАВЛЕНИЯ ПОЛИТИКАМИ КОНСИСТЕНТНОСТИ ДАННЫХ В ГЕОРАСПРЕДЕЛЕННОЙ БИЗНЕС-ИНФОРМАЦИОННОЙ СИСТЕМЕ

Синявина А. П., Макушин А. В.

*Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО», Санкт-Петербург,
Российская Федерация, e-mail: sinyavina.alisa2003@mail.ru*

В статье рассматривается проблема управления согласованностью данных в географически распределенных бизнес-информационных системах, где распределение данных между территориально удаленными узлами повышает доступность и масштабируемость, но усложняет поддержание согласованного состояния из-за сетевых задержек, лага репликации, частичных отказов и конкурирующих изменений. Цель работы заключается в разработке проектных решений для программного компонента управления политиками согласованности данных. В работе применены методы сравнительного анализа, процессного и архитектурного моделирования. Были рассмотрены настраиваемые уровни согласованности, транзакционная согласованность, контекстно зависимое управление согласованностью и централизованное описание правил. На основании анализа сформированы группы требований к компоненту, построена процессная модель выбора политики согласованности при выполнении операции с данными и разработана программная архитектура компонента. Предложенная архитектура предусматривает выделение контура управления политиками и контура применения политики, а также отделяет правила выбора согласованности от прикладной логики и особенностей конкретного хранилища данных. Результаты работы могут быть использованы при разработке прототипа компонента и последующей проверке его применимости в географически распределенной информационной системе в условиях изменяющихся требований к согласованности корпоративных данных.

Ключевые слова: географически распределенная информационная система, согласованность данных, распределенное хранилище, политика согласованности, репликация данных, управление правилами

DESIGN OF A SOFTWARE COMPONENT FOR MANAGING DATA CONSISTENCY POLICIES IN A GEO-DISTRIBUTED BUSINESS INFORMATION SYSTEM

Sinyavina A. P., Makushin A. V.

*Federal State Autonomous Educational Institution of Higher Education
“National Research University ITMO”, Saint Petersburg, Russian Federation,
e-mail: sinyavina.alisa2003@mail.ru*

This article examines the problem of data consistency management in geographically distributed business information systems, where the distribution of data across geographically dispersed nodes enhances availability and scalability, but complicates the maintenance of a consistent state due to network delays, replication lag, partial failures and competing changes. The aim of this work is to develop design solutions for a software component to manage data consistency policies. The work employs methods of comparative analysis, process modelling and architectural modelling. Configurable consistency levels, transactional consistency, context-dependent consistency management and a centralised description of rules were examined. Based on the analysis, groups of requirements for the component were formulated, a process model for selecting a consistency policy when performing data operations was constructed, and the software architecture of the component was developed. The proposed architecture provides for the separation of the policy management loop and the policy application loop, and also separates consistency selection rules from application logic and the specific features of a particular data store. The results of this work can be used in the development of a component prototype and the subsequent verification of its applicability in a geographically distributed information system under conditions of changing requirements for the consistency of corporate data.

Keywords: geographically distributed information system, data consistency, consistency policy, distributed storage, data replication, policy management

Введение

Географически распределенная бизнес-информационная система характеризуется размещением данных и вычислительных компонентов в нескольких территориально удаленных узлах. Такая архитектура позволяет снижать задержки доступа, повышать отказоустойчивость и обеспечивать масштабирование, однако усложняет поддержание согласован-

ного состояния данных из-за сетевых задержек, лага репликации, частичных отказов и конкурирующих изменений.

Для обеспечения согласованности данных используются разные подходы: настраиваемые уровни чтения и записи, транзакционные механизмы распределенных СУБД, ограничения устарелости данных и контекстно зависимый выбор уровня кон-

систентности. При этом единая стратегия консистентности не может быть одинаково эффективной для всех данных и операций: строгая согласованность повышает стоимость межрегиональной координации, а ослабленные модели могут приводить к устаревшим чтениям и нарушению бизнес-инвариантов.

В связи с этим возникает необходимость формализованного управления правилами выбора консистентности с учетом домена данных, типа операции, допустимой устарелости информации и возможностей используемого хранилища. Одним из способов решения данной задачи является проектирование программного компонента управления политиками консистентности данных.

Цель исследования заключается в разработке проектных решений для программного компонента управления политиками консистентности данных в геораспределенной бизнес-информационной системе.

Материалы и методы исследования

Исследование опирается на научные публикации и технические работы, посвященные управлению консистентностью данных в распределенных и геораспределенных информационных системах [1-3]. При анализе также учитывались работы, раскрывающие настраиваемую причинную консистентность и влияние уровня согласованности на задержку выполнения операций [4; 5]. Отдельно были рассмотрены подходы, связанные с реплицируемыми типами данных и проверкой их согласованности [6; 7].

В рамках анализа были рассмотрены четыре группы подходов. Первая группа связана с использованием настраиваемых уровней консистентности при выполнении операций чтения и записи. Данный подход применяется в распределенных хранилищах и позволяет выбирать компромисс между актуальностью данных, доступностью

и задержкой выполнения операции [8-10]. Вторая группа подходов связана с распределенными SQL-СУБД, где согласованность обеспечивается за счет транзакционных механизмов, уровней изоляции и координации между узлами [11-13]. Третья группа подходов основана на выборе уровня консистентности с учетом контекста операции: типа данных, характеристик запроса, допустимой устарелости и состояния системы [4; 5]. При этом влияние выбранного уровня консистентности на производительность особенно важно для NoSQL-хранилищ, где изменение гарантий согласованности может отражаться на задержке и скорости доступа к данным [14]. Четвертая группа подходов связана с вынесением правил принятия решений из прикладного кода в отдельный управляющий слой [15].

Сравнение рассмотренных подходов приведено в таблице 1.

В качестве метода исследования был использован сравнительный анализ. Рассмотренные подходы сопоставлялись по следующим критериям: уровень применения политики, гранулярность настройки, зависимость от конкретной базы данных, поддержка геораспределенности, влияние на производительность, надежность, наблюдаемость и сложность внедрения. Наиболее значимыми для дальнейшего проектирования были определены гранулярность настройки, зависимость от конкретной базы данных, поддержка геораспределенности и влияние выбранного уровня консистентности на задержку выполнения операций. Сопоставление подходов также позволило выделить основные ограничения: отсутствие универсального механизма консистентности для всех сценариев, зависимость реализации от возможностей конкретного хранилища, рост задержек при усилении гарантий согласованности и необходимость формального описания контекста операции.

Таблица 1

Сравнение подходов к управлению консистентностью данных

Подход	Основная идея	Ограничения
Настраиваемые уровни консистентности	Выбор уровня согласованности для операций чтения и записи	Зависимость от механизмов конкретного хранилища
Транзакционная консистентность	Обеспечение согласованности через транзакции, уровни изоляции и координацию узлов	Рост задержек и необходимость обработки конфликтов
Контекстно зависимое управление консистентностью	Выбор уровня консистентности с учетом параметров операции и состояния системы	Необходимость формального описания контекста и правил выбора
Централизованное описание правил	Вынесение правил принятия решений из прикладной логики в отдельный управляющий слой	Не является самостоятельным механизмом обеспечения консистентности

Примечание: составлено авторами на основе источников [4; 5; 8-15].

Эти выводы использованы далее при формировании групп требований и разработке программной архитектуры компонента.

Результаты исследования и их обсуждение

В результате анализа существующих подходов было установлено, что управление консистентностью данных в геораспределенной информационной системе не может быть сведено к единому техническому механизму. Настраиваемые уровни консистентности позволяют задавать разные гарантии для операций чтения и записи, однако зависят от возможностей конкретного хранилища [8; 10]. Транзакционные механизмы распределенных SQL-СУБД обеспечивают строгую согласованность, но требуют межузловой координации и могут увеличивать задержку выполнения операций [11-13]. Подходы с настраиваемыми гарантиями согласованности позволяют выбирать модель консистентности с учетом требований операции, но требуют формального описания параметров такого выбора [4; 5]. Policy-based подходы позволяют вынести правила принятия решений из прикладного кода, но сами по себе не обеспечивают консистентность данных [15].

На основании проведенного анализа были сформированы группы требований к программному компоненту управления политиками консистентности данных (табл. 2). Требования отражают необходимость централизованного описания политик, выбора применимой политики по контексту операции и сопоставления логического требования к согласованности с доступным техническим механизмом реализации.

Для уточнения логики применения компонента была построена процессная модель

выбора политики консистентности при выполнении операции с данными. Данный процесс отражает основной сценарий работы компонента и показывает, какие действия должны быть поддержаны его программной архитектурой.

Процесс начинается с формирования бизнес-сервисом контекста операции. В состав контекста могут входить домен данных, тип операции, регион выполнения, критичность операции и требования к свежести данных. На основании полученного контекста компонент управления политиками выполняет поиск применимых политик и проверяет условия их применения. Если подходящая политика не найдена, компонент формирует отказ в выполнении операции. Если политика найдена, она сопоставляется с доступным механизмом технической реализации, после чего операция передается в слой доступа к данным с учетом выбранных параметров. Представленная процессная модель показывает, что компонент должен поддерживать прием контекста операции, выбор применимой политики, проверку возможности ее реализации и передачу параметров выполнения в слой доступа к данным. Эти действия были использованы как основание для определения состава модулей программной архитектуры.

С учетом сформированных требований и процессной модели была разработана программная архитектура компонента управления политиками консистентности. Компонент рассматривается как отдельный управляющий слой, расположенный между бизнес-сервисами и слоем доступа к данным. Его назначение состоит в отделении правил выбора политики консистентности от прикладной логики и конкретных механизмов хранилищ данных.

Таблица 2

Группы требований к программному компоненту

Группа требований	Содержание
Управление политиками	Создание, просмотр, изменение, удаление и проверка политик консистентности данных
Описание области применения	Задание домена данных, типа операции, региона и иных параметров контекста
Описание требований к консистентности	Задание уровня согласованности, допустимой устарелости данных, требований к порядку операций и видимости изменений
Выбор политики	Получение контекста операции и поиск применимой политики с учетом условий ее применения
Сопоставление с техническим механизмом	Определение способа реализации логической политики средствами выбранного хранилища данных
Обработка исключительных ситуаций	Формирование отказа при отсутствии подходящей политики или неподдерживаемом механизме реализации
Нефункциональные требования	Минимальная дополнительная задержка, доступность, расширяемость, наблюдаемость и независимость логики выбора политики от конкретной базы данных

Примечание: составлено авторами на основе полученных данных в ходе исследования.

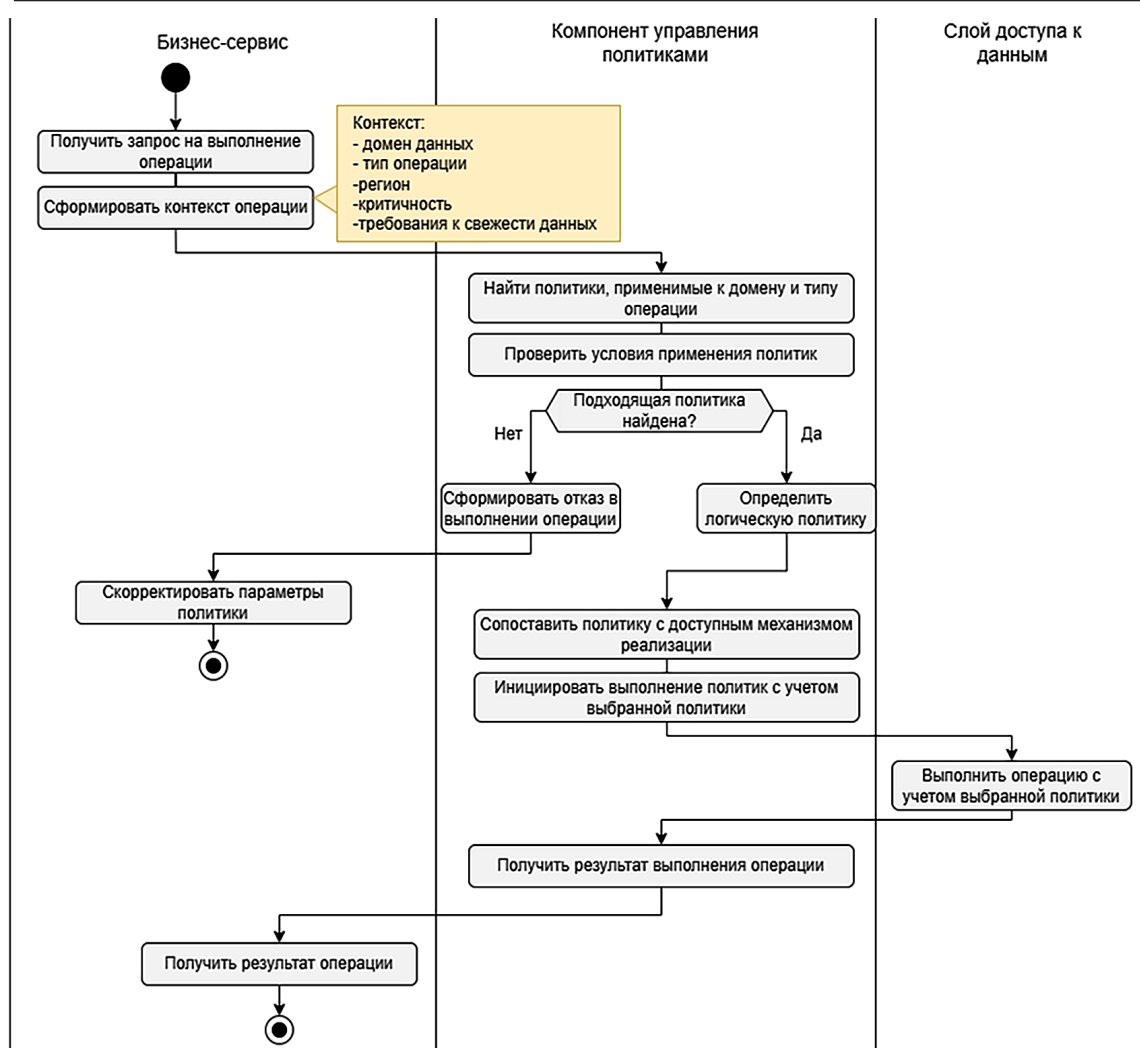


Рис. 1. Процесс выбора политики консистентности при выполнении операции
Источник: составлено авторами по результатам данного исследования

Программная архитектура компонента представлена на рис. 2.

В архитектуре выделены два основных контура. Первый контур связан с управлением политиками и используется администратором для создания и изменения правил консистентности. В него входят API управления политиками, модуль проверки политик и хранилище политик. API управления политиками принимает описание политики, модуль проверки контролирует полноту, корректность и непротиворечивость заданных параметров, после чего корректная политика сохраняется в хранилище.

Второй контур связан с применением политики при выполнении операции с данными. Бизнес-сервис передает в компонент контекст операции, включающий домен данных, тип операции, регион выполнения, критичность операции и требования к свежести данных. API применения политики

передает полученный контекст в модуль выбора политики. Данный модуль выполняет поиск подходящего правила в хранилище политик и проверяет условия его применения. Если политика не найдена, компонент формирует отказ в выполнении операции. Если политика найдена, она передается в модуль сопоставления с механизмом реализации.

Модуль сопоставления с механизмом реализации определяет, каким техническим способом может быть применена выбранная логическая политика. Для этого используется конфигурация механизмов реализации, содержащая сведения о поддерживаемых способах обеспечения консистентности в используемой технологической среде. К таким способам могут относиться уровни согласованности запросов, ограничения устарелости данных и кворумные механизмы [8-10].

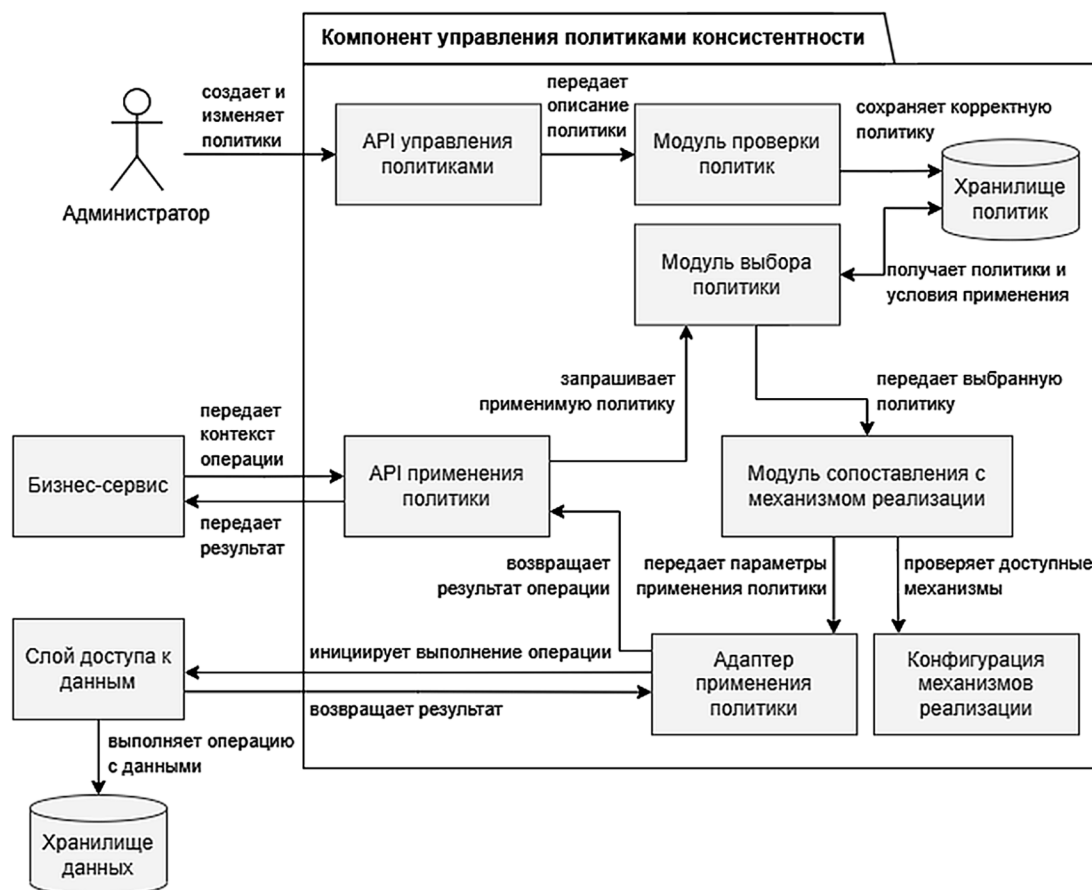


Рис. 2. Программная архитектура компонента управления политиками консистентности данных
 Источник: составлено авторами по результатам данного исследования

Для распределенных SQL-СУБД также могут учитываться транзакционные режимы, уровни изоляции и механизмы координации между узлами [11-13]. При выборе механизма необходимо учитывать его влияние на задержку выполнения операций и производительность доступа к данным [14].

Предложенная архитектура позволяет разделить три уровня ответственности. На прикладном уровне бизнес-сервис формирует контекст операции, но не содержит правил выбора консистентности. На управляющем уровне компонент выбирает применимую политику и определяет способ ее реализации. На технологическом уровне слой доступа к данным и хранилище выполняют операцию с учетом выбранных параметров. Такое разделение снижает зависимость прикладных сервисов от особенностей конкретной СУБД и упрощает изменение правил консистентности при изменении требований к данным.

Таким образом, основным результатом исследования является программная архитектура компонента, обеспечивающего централизованное управление политика-

ми консистентности данных. Компонент не заменяет механизмы распределенных СУБД, а связывает требования к данным и операциям с доступными техническими средствами их реализации. Полученное проектное решение может быть использовано как основа для разработки прототипа и последующей проверки корректности выбора политик в геораспределенной бизнес-информационной системе.

Заключение

Предложенный программный компонент может быть использован в геораспределенных бизнес-информационных системах, где для разных доменов данных и операций требуются различные гарантии консистентности. Его применение позволяет централизовать правила выбора политики, уменьшить зависимость прикладных сервисов от особенностей конкретных хранилищ данных и повысить прозрачность управления требованиями к согласованности. Это особенно важно для систем, в которых необходимо одновременно учитывать актуальность данных, доступность, задержки

выполнения операций и ограничения меж-региональной координации.

Практическая значимость разработанной архитектуры заключается в том, что она создает основу для последующей реализации прототипа и проверки применимости компонента в реальной информационной системе. Ожидаемый эффект от использования такого решения связан с упрощением сопровождения политик консистентности, снижением риска неявного применения разных стратегий в прикладном коде и повышением управляемости работы с данными в геораспределенной среде.

Список литературы

1. Margara A., Cugola G., Felicioni N., Cilloni S. A Model and Survey of Distributed Data-Intensive Systems // *ACM Computing Surveys*. 2023. Vol. 56. № 1. P. 1–69. URL: <https://dl.acm.org/doi/10.1145/3604801> (дата обращения: 15.05.2026). DOI: 10.1145/3604801.
2. Viotti P., Vukolić M. Consistency in Non-Transactional Distributed Storage Systems // *ACM Computing Surveys*. 2016. Vol. 49. № 1. P. 1–34. DOI: 10.1145/2926965.
3. Rambabu D., Govardhan A. Survey on data replication in cloud systems // *Web Intelligence*. 2024. Vol. 22. № 1. P. 83–109. URL: <https://journals.sagepub.com/doi/10.3233/WEB-230087> (дата обращения: 17.05.2026). DOI: 10.3233/WEB-230087.
4. Jiang X., Wei H., Huang Y. Tunable Causal Consistency: Specification and Implementation // *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*. 2022. P. 169–176. URL: <https://ieeexplore.ieee.org/document/10077925> (дата обращения: 17.05.2026). DOI: 10.1109/ICPADS56603.2022.00030.
5. Didona D., Guerraoui R., Wang J., Zwaenepoel W. Causal Consistency and Latency Optimality: Friend or Foe? // *Proceedings of the VLDB Endowment*. 2018. Vol. 11. № 11. P. 1618–1632. URL: <https://vldb.org/pvldb/vol11/p1618-didona.pdf> (дата обращения: 20.05.2026). DOI: 10.14778/3236187.3236210.
6. Preguiça N., Baquero C., Shapiro M. Conflict-Free Replicated Data Types (CRDTs) // *Encyclopedia of Big Data Technologies*. 2022. P. 1–10. URL: https://link.springer.com/referenceworkentry/10.1007/978-3-319-63962-8_185-2 (дата обращения: 20.05.2026). DOI: 10.1007/978-3-319-63962-8_185-2.
7. Biswas R., Emmi M., Enea C. On the Complexity of Checking Consistency for Replicated Data Types // *Computer Aided Verification. CAV 2019. Lecture Notes in Computer Science*. 2019. Vol. 11562. P. 324–343. URL: https://link.springer.com/chapter/10.1007/978-3-030-25543-5_19 (дата обращения: 20.05.2026). DOI: 10.1007/978-3-030-25543-5_19.
8. Lakshman A., Malik P., Cassandra: A Decentralized Structured Storage System // *ACM SIGOPS Operating Systems Review*. 2010. Vol. 44. № 2. P. 35–40. DOI: 10.1145/1773912.1773922.
9. Hackett A. F., Rowe J., Kuppe M. A. Understanding Inconsistency in Azure Cosmos DB with TLA+ // *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 2023. P. 1–12. URL: <https://ieeexplore.ieee.org/document/10172859> (дата обращения: 22.05.2026). DOI: 10.1109/ICSE-SEIP58684.2023.00006.
10. Elhemali M., Gallagher N., Gordon N., Idziorek J., Krog R., Lazier C., Mo E., Mritunjai A., Perianayagam S., Rath T., Sivasubramanian S., Sorenson J.C. III, Sothikul S., Terry D., Vig A. Amazon DynamoDB: A Scalable, Predictably Performant, and Fully Managed NoSQL Database Service // *Proceedings of the 2022 USENIX Annual Technical Conference*. 2022. P. 1037–1048. URL: <https://www.usenix.org/conference/atc22/presentation/elhemali> (дата обращения: 23.05.2026).
11. Taft R., Sharif I., Matei A., VanBenschoten N., Lewis J., Grieger T., Niemi K., Woods A., Birzin A., Poss R., Bardea P., Ranade A., Darnell B., Gruneir B., Jaffray J., Zhang L., Mattis P. CockroachDB: The Resilient Geo-Distributed SQL Database // *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020. P. 1493–1509. DOI: 10.1145/3318464.3386134.
12. Zhou W., Peng Q., Zhang Z., Zhang Y., Ren Y., Li S., Fu G., Cui Y., Li Q., Wu C., Han S., Wang S., Li G., Yu G. GeoGauss: Strongly Consistent and Light-Coordinated OLTP for Geo-Replicated SQL Database // *Proceedings of the ACM on Management of Data*. 2023. Vol. 1. № 1. Article 62. DOI: 10.1145/3588916.
13. Memarzia P., Zhang H., Ho K., Grosman R., Wang J. GaussDB-Global: A Geographically Distributed Database System // *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 2024. P. 5111–5118. URL: <https://ieeexplore.ieee.org/document/10598023> (дата обращения: 25.05.2026). DOI: 10.1109/ICDE60146.2024.00383.
14. Alflahi A. A. E., Mohammed M. A. Y., Alsammani A. Enhancement of Database Access Performance by Improving Data Consistency in a Non-relational Database System (NoSQL) // *Computational Science and Its Applications – ICCSA 2024 Workshops. Lecture Notes in Computer Science*. 2024. Vol. 14816. P. 194–205. URL: https://link.springer.com/chapter/10.1007/978-3-031-65223-3_13 (дата обращения: 27.05.2026). DOI: 10.1007/978-3-031-65223-3_13.
15. Pallevatta S., Babar M. A. Towards Secure Management of Edge-Cloud IoT Microservices Using Policy as Code // *Software Architecture. ECSA 2024. Lecture Notes in Computer Science*. 2024. Vol. 14889. P. 270–287. URL: https://link.springer.com/chapter/10.1007/978-3-031-70797-1_18 (дата обращения: 28.05.2026). DOI: 10.1007/978-3-031-70797-1_18.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.