



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.

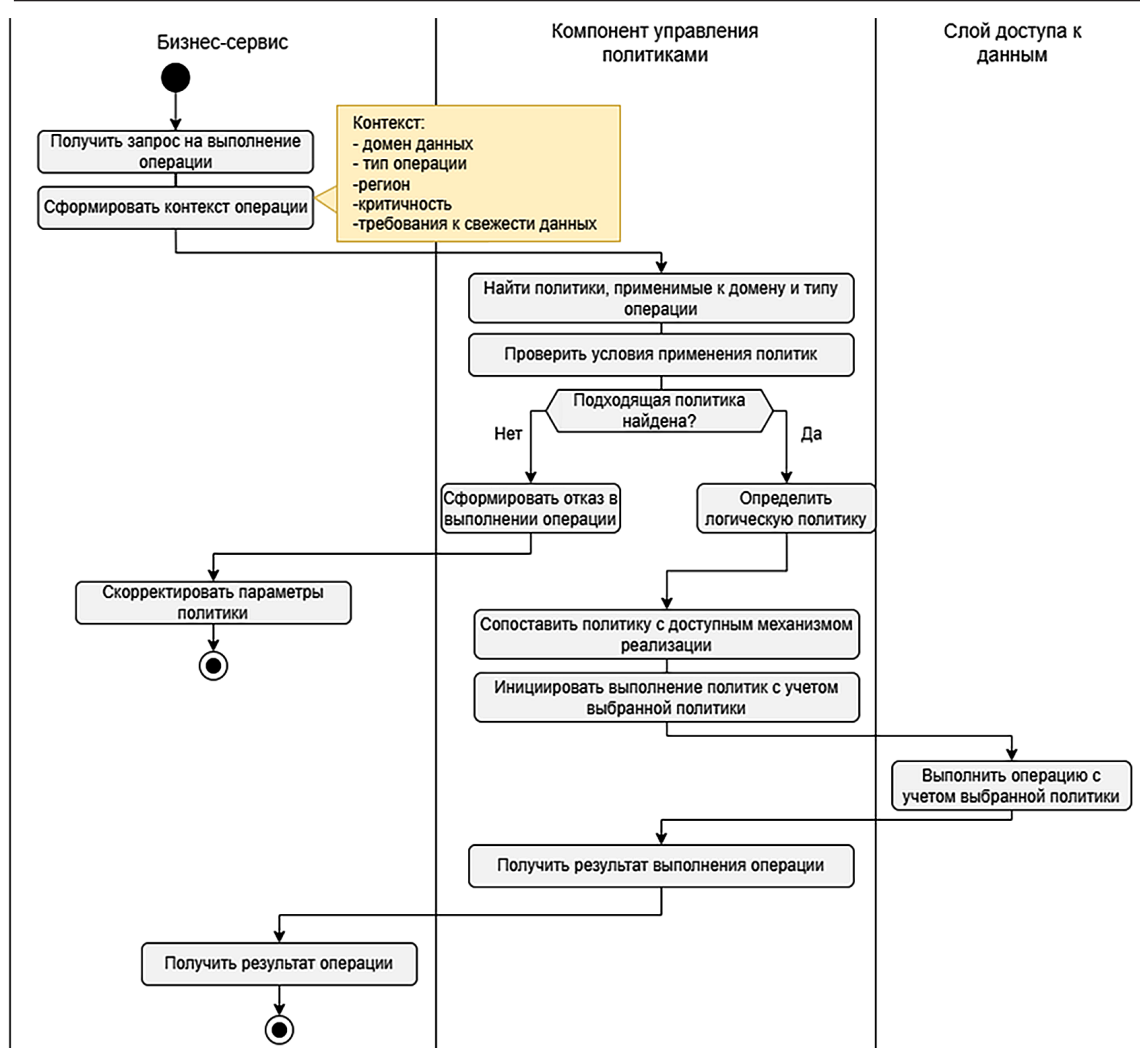


Рис. 1. Процесс выбора политики консистентности при выполнении операции
Источник: составлено авторами по результатам данного исследования

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].

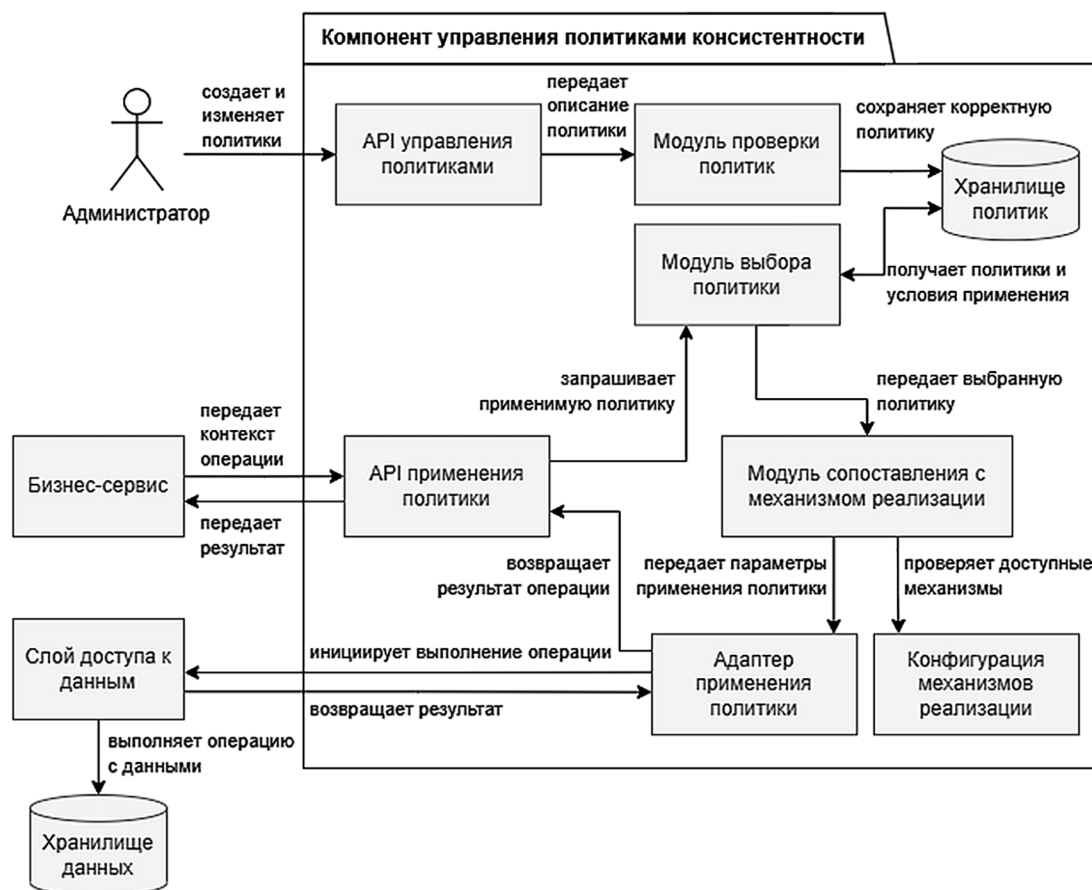


Рис. 2. Программная архитектура компонента управления политиками консистентности данных
 Источник: составлено авторами по результатам данного исследования

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.