

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Черепенин В.А., Смык Н.О., Воробьев С.П. Интеграция облачных, туманных и граничных технологий для оптимизации высоконагруженных систем // Программные системы и вычислительные методы. 2024. № 1. DOI: 10.7256/2454-0714.2024.1.69900 EDN: HYTKBH URL: [https://nbpublish.com/library\\_read\\_article.php?id=69900](https://nbpublish.com/library_read_article.php?id=69900)

## Интеграция облачных, туманных и граничных технологий для оптимизации высоконагруженных систем

**Черепенин Валентин Анатольевич**

ORCID: 0000-0002-6310-1939

аспирант, кафедра Информационные и измерительные системы и технологии, Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова

346428, Россия, Ростовская область, г. Новочеркасск, ул. Просвещения, 132

✉ [cherept2@gmail.com](mailto:cherept2@gmail.com)



**Смык Николай Олегович**

аспирант, кафедра Программное обеспечение вычислительной техники, Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова

346428, Россия, Ростовская область, г. Новочеркасск, ул. Просвещения, 132

✉ [smyk.n@list.ru](mailto:smyk.n@list.ru)



**Воробьев Сергей Петрович**

кандидат технических наук

доцент, кафедра Информационные и измерительные системы и технологии, Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова

346428, Россия, Ростовская область, г. Новочеркасск, ул. Просвещения, 132

✉ [vsp1999@yandex.ru](mailto:vsp1999@yandex.ru)



[Статья из рубрики "Системный анализ, поиск, анализ и фильтрация информации"](#)

**DOI:**

10.7256/2454-0714.2024.1.69900

**EDN:**

HYTKBH

**Дата направления статьи в редакцию:**

13-02-2024

**Дата публикации:**

20-02-2024

**Аннотация:** Исследование посвящено анализу методов и инструментов оптимизации работы высоконагруженных систем с использованием облачных, туманных и граничных технологий. Основное внимание уделяется пониманию концепции высоконагруженных систем, выявлению основных причин увеличения нагрузки на такие системы, а также изучению зависимости нагрузки от уровня масштабирования системы, количества пользователей и объема обрабатываемых данных. Введение этих технологий предполагает создание многоуровневой топологической структуры, которая способствует эффективной работе распределенных корпоративных систем и вычислительных сетей. Рассматриваются современные подходы к управлению нагрузками, исследуются основные факторы, влияющие на производительность, и предлагается модель оптимизации, обеспечивающая высокий уровень эффективности и устойчивости системы к пиковым нагрузкам, обеспечивая при этом непрерывность и качество обслуживания конечных пользователей. Методология основана на комплексном подходе, включающем анализ существующих проблем и предложение новаторских решений для оптимизации, применение архитектурных решений на базе IoT, облачных, туманных и граничных вычислений для улучшения производительности и снижения задержек в высоконагруженных системах. Научная новизна данной работы заключается в разработке уникальной многоуровневой топологической структуры, способной интегрировать облачные, туманные и граничные вычисления для оптимизации высоконагруженных систем. Эта структура позволяет обеспечить улучшенную производительность, снижение задержек и эффективное масштабирование системы, решая при этом проблемы управления большими объемами данных и одновременным обслуживанием множества запросов. Выводы исследования подчеркивают значительный потенциал технологии IoT в улучшении производственных процессов, демонстрируя, как интеграция современных технологических решений может способствовать повышению урожайности, качества продукции и управлению рисками. Результаты работы предоставляют основу для дальнейшего развития умного сельского хозяйства и могут быть применены в различных отраслях для создания эффективных, масштабируемых и унифицированных систем, обеспечивая тем самым новые возможности для устойчивого развития аграрного сектора и других сфер экономики.

**Ключевые слова:**

Высоконагруженные системы, Облачные вычисления, Туманные вычисления, Граничные вычисления, Оптимизация производительности, Масштабируемость, Интернет вещей, Интеграция технологий, Управление данными, Непрерывность сервиса

**Введение**

Термин "высоконагруженная система" обычно применяется к веб-сервисам и сайтам, испытывающим интенсивное взаимодействие с большим количеством пользователей одновременно. Однако это понятие также распространяется на разнообразные информационные системы и приложения для бизнеса, работающие с значительными объемами данных. Поэтому задачи по оптимизации высоконагруженных систем актуальны не только для веб-разработки, но и в контексте любых проектов, предполагающих наличие серверной и клиентской частей. Следовательно, высоконагруженной системой

считается приложение, испытывающее высокую нагрузку из-за множества пользователей, обширных массивов данных или интенсивности вычислений. Эти аспекты могут проявляться как вместе, так и по отдельности, но наличие хотя бы одного из них указывает на повышенные требования к ресурсам системы.

В контексте развития современных дистрибутивных информационных систем, проектирование включает в себя использование преимуществ технологий Интернета вещей (IoT) и применение архитектурных решений, основанных на облачных, туманных и граничных вычислениях. Облачные вычисления представляют собой метод построения распределенной системы с доступом к гибкому и масштабируемому набору ресурсов через интернет. Туманные вычисления функционируют как промежуточный слой между конечными устройствами и центрами данных, сокращая задержки и сетевой трафик. Граничные вычисления дополнительно приближают обработку данных к пользователю, выполняя большинство операций на периферии сети, что способствует мгновенному реагированию на полученные данные благодаря использованию программируемых логических контроллеров для управления процессами [\[1\]](#).

Разработка эффективной архитектуры для такого сложного взаимодействующего комплекса предполагает использование многоуровневого подхода к представлению топологии вычислительной сети, обеспечивая оптимизацию работы системы в целом.

#### Методология исследования

Задачей данной статьи является детальное рассмотрение стратегий и инструментария для повышения эффективности функционирования систем с высоким уровнем нагрузки. Исследуемый объект — это системы, обеспечивающие функциональность веб-сайтов, одновременно обрабатывающие запросы множества пользователей. В качестве методологической базы применяются процедуры анализа и синтеза, а также техники обобщения существующих данных и исследований в данной области [\[2\]](#).

#### Результаты исследования и их обсуждение

Анализируя высоконагруженные системы, ключевое внимание уделяется их специфическим атрибутам:

1. Высоконагруженные системы характеризуются строгой архитектурой с ограниченным пространством для модификаций в отдельных подсистемах. Их сложная внутренняя структура ограничивает возможности для глубокой адаптации, делая систему менее гибкой из-за уникальности каждой конфигурации. Обработка данных в таких системах, требующая высокой стабильности и надежности, предполагает тщательный подбор структур баз данных, исходя из их специфики, объема данных и интенсивности запросов. Попытки увеличить гибкость системы могут привести к значительным затратам ресурсов, поскольку это требует комплексного переосмысления и возможной реорганизации основных принципов ее работы [\[3\]](#).
2. Одним из ключевых атрибутов высоконагруженных систем является их способность обеспечивать мгновенную реакцию. В контексте обработки данных через запросы, быстродействие системы критически важно: задержки в обработке запросов напрямую влияют на время ожидания пользователем необходимой информации.
3. Масштабируемость является ключевым аспектом для систем с высоким уровнем загрузки, поскольку увеличение объема обрабатываемых данных может значительно повысить нагрузку на информационную инфраструктуру. Для адаптации к растущим

требованиям, масштабируемость обычно достигается двумя основными методами. Первый — вертикальное масштабирование, предполагает повышение производительности отдельных элементов системы для усиления её общей мощности. Этот метод не требует структурных изменений в архитектуре и часто реализуется за счёт модернизации оборудования. Второй метод — горизонтальное масштабирование, включает в себя распределение нагрузки между несколькими серверами или узлами, работающими параллельно, что требует добавления новых компонентов в систему и соответствующей настройки программного обеспечения для эффективного взаимодействия между ними. Хотя горизонтальное масштабирование предполагает большую сложность в реализации, оно обеспечивает более гибкое и масштабируемое решение в долгосрочной перспективе и часто выбирается как наиболее предпочтительный вариант [4]. Важно тщательно анализировать потребности системы и определять оптимальный подход к масштабированию, учитывая специфику и требования к производительности.

4. Разработка модульной архитектуры для высоконагруженных систем предусматривает организацию их структуры в виде отдельных компонентов, которые затем интегрируются на разнообразные серверные платформы. Этот подход позволяет распределить часть нагрузки, назначая особо интенсивно используемые модули на множество серверов для их параллельной работы. Хотя такое решение улучшает производительность, оно может привести к проблемам с согласованностью данных из-за их параллельной обработки [5]. С увеличением общей нагрузки на систему усиливается риск возникновения несогласованности данных, вызванной одновременными операциями разных пользователей. Поэтому предпочтительнее выбирать стратегию, при которой интенсивно используемые процессы выделяются на отдельные, более мощные сервера без параллелизации, что способствует повышению общей эффективности и надежности системы.

5. Интенсивная нагрузка на интеграционный уровень системы напрямую связана с её модульной структурой. Расширение системы за счёт добавления новых модулей увеличивает количество взаимодействий между ними, требуя от коммуникационных процессов высокой скорости и надёжности. С увеличением числа модулей сложность этих взаимодействий возрастает, что ведёт к усиленной загрузке на интеграционный слой системы, особенно когда объём коммуникаций увеличивается экспоненциально.

6. Уникальность является ключевым атрибутом высоконагруженных приложений, подчёркивая отсутствие унифицированных стандартных подходов к их разработке. Это означает, что каждое решение разрабатывается с учётом специфических потребностей и требований конкретного бизнеса, делая систему неповторимой и точно адаптированной под задачи заказчика [6].

7. Эффективное применение стратегии редундантности для ключевых элементов системы становится критичным для поддержания непрерывности бизнес-процессов, особенно в контексте высоконагруженных систем. Для гарантии стабильности работы такие системы оборудуются дублирующими узлами, как в программном, так и в аппаратном аспектах. Это не подразумевает постоянную параллельную активность всех резервных компонентов, а скорее предусматривает их готовность к моментальному включению в работу при возникновении критических нагрузок, обеспечивая таким образом разгрузку основной системы и её бесперебойное функционирование.

Высокие нагрузки часто определяются не самой архитектурой системы, а условиями её эксплуатации, отражая специфику деятельности в определённой бизнес-сфере. Среди

причин, влияющих на увеличение нагрузки, можно выделить:

- Увеличение числа запросов к системам управления взаимоотношениями с клиентами (CRM) и планирования ресурсов предприятия (ERP), обслуживающих множество пользователей, а также в поддержке клиентов и контакт-центрах, где происходит обработка значительного количества звонков и обращений;
- Рост объёма обрабатываемой информации в системах мониторинга с большим количеством подключённого оборудования, в инструментах бизнес-аналитики, а также в CRM и ERP системах, обрабатывающих обширные массивы данных;
- Ошибки в настройке системы, часто возникающие из-за недочётов в программном коде или отсутствия должной оптимизации, приводящие к усилению нагрузки на серверные ресурсы.

Для снижения и управления нагрузкой на систему применяются различные стратегии оптимизации, включающие:

- Применение сетевых протоколов и внешних библиотек для уменьшения числа запросов, включая методы кэширования данных, как на уровне запросов к базе данных, так и при получении ответов от сервера, что способствует снижению задержек и увеличению производительности системы;
- Оптимизация работы с базой данных через индексирование, что ускоряет поиск и обработку данных, репликацию для распределения нагрузки и обеспечения отказоустойчивости, а также секционирование для эффективного управления и хранения больших объемов информации [\[7\]](#).

Детализируя методы оптимизации взаимодействия с базой данных, следует выделить несколько ключевых направлений:

1. Сериализация и десериализация данных играют важную роль в процессе обмена информацией между сервером и клиентом. Эти процедуры обеспечивают корректную передачу данных, но также влекут за собой временные затраты. Оптимизация данных процессов позволяет ускорить обработку информации и улучшить общую производительность системы.
2. Кэширование запросов к базе данных становится эффективным решением для снижения нагрузки на информационные хранилища. Организация кэша позволяет временно сохранять результаты часто выполняемых или редко изменяемых запросов, существенно уменьшая количество обращений к базе данных. Для реализации кэширования часто применяются хэш-таблицы, обеспечивающие быстрый доступ к сохраненным данным.
3. Индексирование базы данных представляет собой создание специализированных структур (индексов), которые способствуют ускорению поиска данных. Индексы не только повышают скорость доступа к информации, но и поддерживают целостность данных. Однако, следует учитывать, что операции с индексами, такие как их перестроение после удаления данных, могут потребовать дополнительных ресурсов. Применение индексов оправдано в базах данных с большим объемом записей, где они могут значительно оптимизировать процессы поиска и обработки информации.

Эти методы оптимизации позволяют улучшить производительность высоконагруженных систем, снизить время ответа на запросы и увеличить общую эффективность работы с

данными [\[8\]](#).

Оптимизация доступа к данным в базах данных может включать стратегию репликации, которая служит для увеличения пропускной способности и доступности системы. Репликация достигается путем создания копий информационной базы, разделяя роли между первичными (ведущими) и вторичными (ведомыми) узлами для балансировки нагрузки операций чтения и обеспечения актуальности данных через синхронизацию. Ведущий узел обрабатывает как чтение, так и запись, распространяя обновления по ведомым узлам. Существуют различные методы репликации:

1. Синхронная репликация обеспечивает полную согласованность данных между узлами, требуя подтверждения записи от всех участников, что повышает надежность, но увеличивает время отклика.
2. Асинхронная репликация ускоряет работу системы, переключаясь на следующие операции сразу после записи на ведущем узле без ожидания подтверждения от ведомых, чем жертвует гарантией актуальности данных на всех узлах.
3. Модель "Ведущий-Ведомые" предполагает наличие одного ведущего узла, обрабатывающего все операции, и множества ведомых, синхронизирующихся через фиксированные интервалы. Это упрощает структуру, но создает риски при отказе ведущего узла.
4. Модель "Ведущие-Ведомые" вносит дополнительную гибкость, добавляя несколько ведущих узлов и повышая отказоустойчивость за счет усложнения процесса синхронизации данных между ними.

Каждый из этих подходов имеет свои преимущества и недостатки, выбор метода репликации зависит от специфических требований к производительности, доступности и консистентности данных в конкретной системе.

Оптимизация баз данных через секционирование представляет собой стратегию разделения данных на меньшие сегменты, направленную на улучшение производительности и пропускной способности [\[9\]](#). Этот подход позволяет балансировать нагрузку, предотвращая перегрузку отдельных узлов и способствуя более эффективной обработке данных. Секционирование может осуществляться разными методами:

- По диапазону ключей – это базовый метод, при котором данные распределяются согласно определенным ключам. Главным вызовом здесь является выбор ключа, который обеспечит оптимальное разделение данных.
- С использованием хэш-функций – более продвинутая техника, где специальная хэш-функция равномерно распределяет данные по секциям. Этот метод обеспечивает более сбалансированное распределение данных и эффективность работы системы.

Для поддержания оптимальной производительности системы после секционирования необходимо регулярно проводить ребалансировку [\[10\]](#). Это связано с тем, что запись данных в разные секции может происходить неравномерно, что со временем может привести к замедлению скорости обработки запросов.

## Заключение

В завершении обсуждения стратегий оптимизации важно подчеркнуть, что эффективное повышение производительности системы требует комплексного подхода, включающего

применение множества разнообразных методик. Каждая из них обладает своими уникальными особенностями и предполагает различные уровни воздействия на архитектуру и процессы взаимодействия в системе. Выбор конкретного метода или их комбинации должен базироваться на тщательном анализе текущего состояния системы, определении ключевых точек, требующих оптимизации, а также на учете специфических потребностей и ожиданий заказчика.

Среди возможных направлений для улучшения производительности могут быть выделены:

- Анализ и оптимизация алгоритмов работы с данными, включая пересмотр запросов к базам данных, использование более эффективных алгоритмов сериализации и десериализации данных.
- Кэширование часто запрашиваемой информации для снижения нагрузки на базы данных и ускорения доступа к данным.
- Масштабирование системы, как вертикальное, так и горизонтальное, для обеспечения гибкости и масштабируемости в ответ на растущие нагрузки.
- Репликация и секционирование баз данных для улучшения доступности и распределения нагрузки между серверами.
- Использование современных подходов к разработке, включая микросервисную архитектуру, для повышения модульности и упрощения масштабирования и обновления системы.

Таким образом, стратегия оптимизации должна быть многоаспектной и учитывать как технические возможности существующей инфраструктуры, так и долгосрочные бизнес-цели заказчика. Разработчикам необходимо гибко подходить к выбору инструментов и методик, адаптируя их под конкретные задачи и условия эксплуатации системы, чтобы достичь оптимального соотношения между производительностью, стабильностью и расширяемостью.

## **Библиография**

1. Catal, C.; Tekinerdogan, B. Aligning education for the life sciences domain to support digitalization and Industry 4.0. *Procedia Computer Science*. 2019, 158, 99–106. DOI:10.1016/j.procs.2019.09.032
2. Patel, C.; Doshi, N. A novel MQTT security framework in generic IoT model. *Procedia Computer Science*. 2020, 171, 1399–1408. DOI:10.1016/j.procs.2020.04.150
3. Subeesh, A.; Mehta, C.R. Automation and digitization of agriculture using artificial intelligence and internet of things. *Artificial Intelligence in Agriculture*. 2021, 5, 278–291. DOI:10.1016/j.aiia.2021.11.004
4. Faridi, F.; Sarwar, H.; Ahtisham, M.; Kumar, S.; Jamal, K. Cloud computing approaches in health care. *Materials Today: Proceedings*, 2022, 51, 1217–1223. DOI:10.1016/j.matpr.2021.07.210
5. Tzounis, A.; Katsoulas, N.; Bartzanas, T.; Kittas, C. Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering*. 2017, 164, 31–48. DOI:10.1016/j.biosystemseng.2017.09.007
6. Tao, W.; Zhao, L.; Wang, G.; Liang, R. Review of the internet of things communication technologies in smart agriculture and challenges. *Computers and Electronics in Agriculture*. 2021, 189, 106352. DOI:10.1016/j.compag.2021.106352

7. Moysiadis, V.; Sarigiannidis, P.; Vitsas, V.; Khelifi, A. Smart farming in Europe. Computer Science Review. 2021, 39, 100345. DOI:10.1016/j.cosrev.2020.100345
8. Raj, M.; Gupta, S.; Chamola, V.; Elhence, A.; Garg, T.; Atiquzzaman, M.; Niyato, D. A survey on the role of Internet of Things for adopting and promoting Agriculture 4.0. Journal of Network and Computer Applications. 2021, 187, 103107. DOI:10.1016/j.jnca.2021.103107
9. Boursianis, A.D.; Papadopoulou, M.S.; Diamantoulakis, P.; Liopa-Tsakalidi, A.; Barouchas, P.; Salahas, G.; Karagiannidis, G.; Wan, S.; Goudos, S.K. Internet of Things (IoT) and agricultural unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review. Internet of Things. 2022, 18, 100187. DOI:10.1016/j.iot.2020.100187
10. Singh, S.; Chana, I.; Buyya, R. Agri-Info: Cloud based autonomic system for delivering agriculture as a service. Internet of Things. 2020, 9, 100131. DOI:10.1016/j.iot.2019.10013

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Рецензируемая работа посвящена исследованию интеграции облачных, туманных и граничных технологий для оптимизации высоконагруженных систем.

Методология исследования базируется на изучении и обобщении научных публикаций по рассматриваемой теме, применении процедур анализа и синтеза, а также техники обобщения существующих данных и исследований в данной области.

Актуальность работы авторы связывают с тем, что задачи оптимизации высоконагруженных систем актуальны не только для веб-разработки, но и в контексте любых проектов, предполагающих наличие серверной и клиентской частей.

Научная новизна рецензируемого исследования, по мнению рецензента, состоит в обобщении стратегий и инструментария для повышения эффективности функционирования систем с высоким уровнем нагрузки.

В тексте статьи выделены следующие разделы: Введение, Методология исследования, Результаты исследования и их обсуждение, Заключение, Библиография.

В статье высоконагруженной системой считается приложение, испытывающее высокую нагрузку из-за множества пользователей, обширных массивов данных или интенсивности вычислений. Особое внимание авторы уделяют специфическим атрибутам высоконагруженных систем: строгой архитектуре с ограниченным пространством для модификаций в отдельных подсистемах; способности обеспечивать мгновенную реакцию; масштабируемости; организация их структуры в виде отдельных компонентов, которые затем интегрируются на разнообразные серверные платформы; модульная структура, уникальность, применение стратегии редундантности для ключевых элементов системы. В публикации выделены причины, влияющие на увеличение нагрузки системы; названы применяемые для управления нагрузкой стратегии оптимизации; выделены ключевые направлений методов оптимизации, которые позволяют улучшить производительность высоконагруженных систем, снизить время ответа на запросы и увеличить общую эффективность работы с данными; рассмотрены подходы к репликации в зависимости от специфических требований к производительности, доступности и консистентности данных в конкретной системе. В Заключении отмечено, что эффективное повышение производительности системы требует комплексного подхода, отражены возможные



направления для улучшения производительности, достижения оптимального соотношения между производительностью, стабильностью и расширяемостью системы.

Библиографический список включает 10 источников – научные публикации на английском языке по рассматриваемой теме, на которые в тексте приведены адресные ссылки, что подтверждает наличие апелляции к оппонентам.

В качестве замечаний, следует отметить отсутствие ссылок на научные работы, опубликованные на русском языке – представляется, что существуют и русскоязычные публикации, заслуживающие внимания; в тексте встречаются несогласованные предложения, например, «Анализируя высоконагруженные системы, ключевое внимание уделяется их специфическим атрибутам...» и др.

Рецензируемый материал соответствует направлению журнала «Программные системы и вычислительные методы», отражает результаты проведенной авторами работы, может вызвать интерес у читателей, поскольку содержит интересные сведения об интеграции облачных, туманных и граничных технологий для оптимизации высоконагруженных систем.