

Багмут Алина Васильевна, магистр  
Кубанский государственный технологический университет

## АРХИТЕКТУРНЫЕ ПОДХОДЫ И ТЕХНОЛОГИЧЕСКИЕ РЕШЕНИЯ ПРИ РАЗРАБОТКЕ МАСШТАБИРУЕМЫХ ВЕБ-ПРИЛОЖЕНИЙ

**Аннотация.** Статья посвящена анализу архитектурных подходов и технологических решений, обеспечивающих масштабируемость современных веб-приложений. Рассматриваются ключевые принципы проектирования – микросервисная архитектура, событийно-ориентированные модели и асинхронные механизмы взаимодействия, формирующие гибкость и распределённость системы. Особое внимание уделяется технологической инфраструктуре, включающей контейнеризацию, оркестрацию, распределённые базы данных, брокеры сообщений и облачные платформы, которые обеспечивают автоматическое управление ресурсами, отказоустойчивость и стабильность приложения при росте нагрузки. Результаты исследования показывают, что масштабируемость достигается благодаря согласованному применению архитектурных и технологических решений, обеспечивающих устойчивую производительность и адаптивность веб-приложений

**Ключевые слова:** Масштабируемость веб-приложений, микросервисная архитектура, асинхронное взаимодействие, брокеры сообщений, распределённые базы данных, облачные технологии

Стремительный рост цифровых сервисов, увеличение объёмов пользовательского трафика и усложнение бизнес-логики современного программного обеспечения формируют устойчивый спрос на веб-приложения, способные эффективно масштабироваться при изменяющихся нагрузках. Масштабируемость становится не только эксплуатационной характеристикой, но и ключевым критерием архитектурного качества системы, определяя её способность обеспечивать высокую доступность, предсказуемую производительность и устойчивость к отказам в условиях динамичных и распределённых сред. Современные веб-приложения функционируют в экосистемах, включающих облачные платформы, контейнерные оркестраторы, распределённые базы данных, брокеры сообщений и системы непрерывной доставки. В этих условиях архитектурные решения должны учитывать требования к горизонтальному масштабированию, изоляции компонентов, обработке высоких объёмов параллельных запросов, а также к управлению состоянием в распределённых средах. Особое значение приобретают подходы, основанные на микросервисной архитектуре, событийно-ориентированных моделях, асинхронных механизмах взаимодействия и многоуровневой балансировке нагрузки [1].

Цель исследования – проанализировать архитектурные подходы и технологические решения, обеспечивающие масштабируемость веб-приложений, а также систематизировать ключевые принципы построения отказоустойчивых и высокопроизводительных систем, применимых в современных условиях разработки и эксплуатации.

### Архитектурные принципы, определяющие масштабируемость веб-приложений

Масштабируемость веб-приложений напрямую зависит от выбранных архитектурных подходов и механизмов взаимодействия между компонентами. Современные нагрузки – от интенсивных пользовательских запросов до высокочастотных событийных потоков – требуют не только вертикального наращивания ресурсов, но и гибких распределённых моделей [2, 3]. Микросервисная архитектура обеспечивает независимость функциональных модулей и возможность масштабировать отдельные сервисы без изменения общей структуры системы. Событийно-ориентированные модели позволяют организовывать асинхронную коммуникацию между компонентами, снижая связанность и повышая устойчивость к пиковым нагрузкам [4].

Асинхронные методы обмена данными увеличивают пропускную способность системы, так как позволяют обслуживать множество параллельных запросов без



блокирующих операций. Балансировка нагрузки распределяет входящий трафик между узлами, предотвращая локальные перегрузки и обеспечивая равномерное использование ресурсов [5]. Совокупность этих подходов формирует архитектуру, способную поддерживать высокую доступность и гарантировать эффективную адаптацию к увеличению числа пользователей. Схематическое представление ключевых архитектурных решений приведено на рисунке 1.

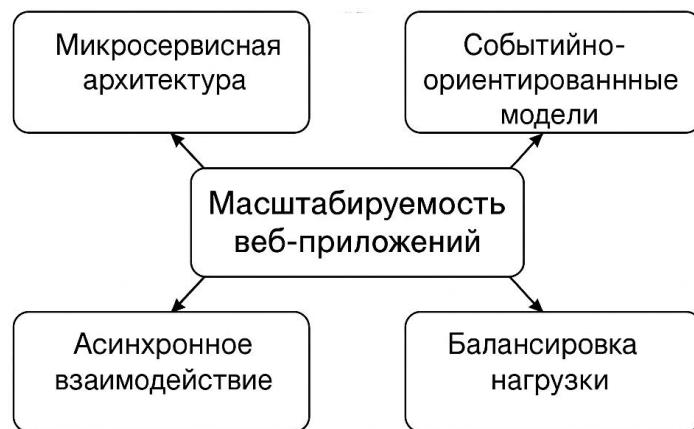


Рисунок 1. Архитектурные подходы к обеспечению масштабируемости веб-приложений

Представленная схема показывает, что масштабируемость достигается благодаря сочетанию нескольких архитектурных решений: модульности микросервисов, событийной организации логики, асинхронного обмена данными и балансировки нагрузки. Эти принципы работают совместно и формируют устойчивую архитектуру, способную адаптироваться к росту трафика и поддерживать высокую производительность в условиях распределённой среды [6].

#### Технологические решения, обеспечивающие масштабируемость и устойчивость веб-приложений

Выбор технологического стека оказывает определяющее влияние на способность веб-приложения поддерживать стабильную производительность при росте нагрузки [7]. В современных распределённых системах ключевую роль играют механизмы автоматизации, контейнеризации и оркестрации, позволяющие гибко управлять ресурсами и быстро адаптироваться к изменению пользовательского трафика. Контейнерные платформы обеспечивают изоляцию сервисов, воспроизводимость окружения и удобство развертывания, а системы оркестрации координируют масштабирование, обновление и восстановление компонентов [8, 9]. Наряду с этим критически важными становятся подходы к хранению данных, поддерживающие горизонтальное масштабирование, а также асинхронные коммуникационные механизмы, снижающие связанность между сервисами.

Облачные вычислительные платформы предоставляют дополнительные возможности для масштабирования за счёт автоматического выделения ресурсов, гибких конфигураций и распределённых систем доставки контента. Использование брокеров сообщений упрощает координацию событий, сокращает нагрузку на серверы и позволяет перераспределять вычислительные задачи в зависимости от текущих условий [10]. Взаимодействие технологических решений формирует основу устойчивой инфраструктуры, способной выдерживать высокие пиковые нагрузки и обеспечивать предсказуемое время отклика даже в условиях интенсивной параллельной обработки запросов [11]. Краткая характеристика ключевых технологий представлена в таблице 1.



Таблица 1. Ключевые технологические решения для разработки масштабируемых веб-приложений

Технология	Значение для масштабируемости	Особенности применения
<b>Контейнеризация (Docker, Podman)</b>	Обеспечивает изоляцию сервисов и воспроизводимость окружения	Ускоряет CI/CD, снижает зависимость от инфраструктуры
<b>Оркестраторы (Kubernetes, Nomad)</b>	Автоматическое масштабирование и управление ресурсами	Поддержка self-healing, распределённый контроль состояния
<b>Брокеры сообщений (Kafka, RabbitMQ)</b>	Снижение связанности между сервисами, асинхронная обработка	Высокая пропускная способность, гибкая маршрутизация событий
<b>Распределённые базы данных (Cassandra, CockroachDB)</b>	Горизонтальное масштабирование и отказоустойчивость	Поддержка шардирования и репликации
<b>CDN-сети доставки контента</b>	Снижение латентности и нагрузки на сервер	Геораспределённое кэширование статического контента
<b>Облачные платформы (AWS, GCP, Azure)</b>	Автоматическое выделение ресурсов и управление инфраструктурой	Поддержка serverless-моделей, глобальных сетей и managed-сервисов

Представленные технологические решения демонстрируют, что масштабируемость веб-приложений обеспечивается сочетанием изоляции сервисов, автоматизации управления ресурсами, асинхронного обмена данными и распределённого хранения информации [12]. Контейнеризация, оркестрация, брокеры сообщений и облачные платформы формируют взаимодополняющую инфраструктуру, которая позволяет системам адаптироваться к росту нагрузки, поддерживать высокую доступность и снижать риск отказов. Таблица подчёркивает, что ключевую роль играет не отдельная технология, а согласованное использование целого комплекса решений.

### Заключение

Исследование показало, что масштабируемость веб-приложений формируется через сочетание архитектурных принципов и технологических решений, обеспечивающих гибкость, распределённость и устойчивость системы. Микросервисная архитектура, событийно-ориентированные модели и асинхронные механизмы взаимодействия позволяют масштабировать отдельные компоненты, повышая пропускную способность и снижая связанность.

Контейнеризация, оркестрация, брокеры сообщений, распределённые базы данных и облачные платформы создают инфраструктуру, поддерживающую автоматическое управление ресурсами и высокую отказоустойчивость. В совокупности эти решения обеспечивают способность веб-приложений адаптироваться к росту нагрузки и поддерживать стабильную производительность. Эффективная масштабируемость достигается не отдельной технологией, а их согласованным применением.

### Список литературы:

1. Фешина Е.В., Юрченко А.А., Махлужев Д.А. Экономическое обоснование выбора архитектурных решений при проектировании веб-приложений // Естественно-гуманитарные исследования. 2024. № 4 (54). С. 262-264.



2. Topalidi A. Exploring architectural patterns for modular web applications: the Rails Engines approach to business logic isolation // Cold Science. 2025. № 13. P. 18-26.
3. Shethiya A.S. Scalability and Performance Optimization in Web Application Development // Integrated Journal of Science and Technology. 2025. Vol. 2. № 1.
4. Garifullin R. Progressive Web Applications (PWA): combining the advantages of the web and native apps // International Journal of Advances in Computer Science and Technology. 2025. Vol. 14 (4). P. 205-208.
5. Zhaksylykov N. Digitalization of project management in the IT sector: case studies and models // Professional Bulletin: Economics and Management. 2024. № 4/2024. P. 50-54.
6. Guttha P.R. Architecting Scalable Business Applications: Design, Development, and Delivery Strategies // Australian Journal of Cross-Disciplinary Innovation. 2023. Vol. 5. № 5.
7. Berezhnoy A. Architectural design patterns for high-load systems: principles, tools, and scalability constraints // Professional Bulletin. Information Technology and Security. 2025. № 3/2025. P. 33-39.
8. Velepucha V., Flores P. A survey on microservices architecture: Principles, patterns and migration challenges // IEEE access. 2023. Vol. 11. P. 88339-88358.
9. Калашников Н.А. Применение архитектурных паттернов в построении чистой архитектуры веб-приложений // Universum: технические науки. 2024. Т. 1. № 12 (129). С. 37-42.
10. Li Z., Guo L., Cheng J., Chen Q., He B., Guo M. The serverless computing survey: A technical primer for design architecture // ACM Computing Surveys (CSUR). 2022. Vol. 54. № 10s. P. 1-34.
11. Pappula K.K. Architectural Evolution: Transitioning from Monoliths to Service-Oriented Systems // International Journal of Emerging Research in Engineering and Technology. 2022. Vol. 3. № 4. P. 53-62.
12. Matias M., Ferreira E., Mateus-Coelho N., Ferreira L. Enhancing Effectiveness and Security in Microservices Architecture // Procedia Computer Science. 2024. Vol. 239. P. 2260-2269.

