

КРИТЕРІЇ ВИБОРУ АРХІТЕКТУР ВЕБ-ЗАСТОСУНКІВ З ВИСОКОЮ ДОСТУПНІСТЮ

Горбатенко М.С., Білова Т.Г.

e-mail: mariia.horbatenko@nure.ua, tetiana.bilova@nure.ua

Харківський національний університет радіоелектроніки, каф. СТ
м. Харків, Україна

In today's world of technology, choosing a high availability web application architecture is critical due to the widespread use of such systems in government agencies, educational institutions, and businesses. The uninterrupted operation of applications affects financial performance, customer confidence, and competitiveness. High availability, fault tolerance, and performance determine the success of the service, as users prefer stable and fast solutions. With the development of IT technologies, new approaches to creating reliable systems are emerging, but choosing the optimal architecture requires careful analysis and consideration of project specifics.

У сучасному світі технологій важливим питанням є вибір архітектур веб-застосунків з високою доступністю через декілька ключових факторів. Наразі багато державних установ, освітніх закладів, таких як школи, університети, приватні освітні заклади, та різноманітні компанії використовують веб-застосунки для надання своїх послуг, комунікацій з клієнтами та обробки даних. Доступ до системи стає критично важливим для користувачів, бо через його втрату можна понести фінансові збитки та втратити клієнтів. Користувачі очікують безперебійної та стабільної роботи веб-застосунків, тому треба враховувати відмовостійкість, масштабованість та продуктивність системи. Це все створює велику конкуренцію, тому перевагу надають веб-застосунку, у якому швидкість реакції на запити вища та забезпечена стабільна та безперебійна робота, аніж сервісу з низькою доступністю. Ринок ІТ-технологій розвивається і з'являються нові можливості для створення систем з високою доступністю, однак для вибору правильної архітектури веб-застосунку потрібен глибокий аналіз та розуміння потреб системи. Таким чином, вибір правильної архітектури веб-застосунків із високою доступністю забезпечує стабільність системи та сприяє її ефективності, конкурентоспроможності та довгостроковій успішності.

Висока доступність (High Availability, HA) – це властивість системи залишатися працездатною та доступною для користувачів упродовж максимально можливого часу. Основна мета високої доступності – забезпечити безперервний доступ до веб-застосунку навіть у разі збоїв окремих компонентів системи. Для оцінки високої доступності потрібно враховувати такі метрики, як угода про рівень обслуговування, яка визначає гарантований час доступності системи (Service Level Agreement); відсоток часу, протягом якого система працює безперебійно (Uptime); середній час між відмовами (Mean Time Between Failures); середній час відновлення після збою (Mean Time to Repair); стійкість системи до відмов,

здатність працювати при виході з ладу одного або кількох компонентів (Fault Tolerance) [1].

Існує декілька практичних прикладів, де висока доступність веб-сервісів критично важлива, таких, як збій у системи онлайн-магазину, який може призвести до втрати продажів та зниження довіри клієнтів; відмова системи на стрімінговій платформі під час трансляції може призвести до втрати аудиторії та репутації; банківські системи повинні буди завжди доступні для здійснення платежів або переказів; недоступність до медичних інформаційних систем може загрожувати життю пацієнтів.

Основні типи архітектур веб-застосунків: монолітна, мікросервісна та безсерверна (serverless) архітектури.

Монолітна архітектура – єдина кодова база, де всі компоненти взаємопов'язані. Їй притаманні проста розробка, легкість розгортання та тестування, але також є недоліки, такі як складність масштабування, висока вразливість до збоїв та низька гнучкість [2].

Мікросервісна архітектура включає в себе окремі сервіси, які можуть працювати незалежно. Це означає, що вихід з ладу одного сервісу не порушить роботу всього веб-додатку, тобто забезпечено високу масштабованість та стійкість до відмов. До недоліків такої системи відносять складність управління, високі вимоги до мережевої взаємодії та потребу у належному керуванні [3].

Безсерверна архітектура орієнтується на автоматичне масштабування та події, усуваючи необхідність у керуванні серверами, що забезпечує високу доступність та знижує витрати на інфраструктуру. Але така система має залежність від постачальника хмарних послуг та обмежений контроль над виконуваним середовищем [4].

Отже, монолітна архітектура вразлива до збоїв та складно масштабується, мікросервісна архітектура має високу доступність та незалежність серверів один від одного, безсерверна архітектура автоматично адаптується до навантажень та дозволяє мінімізувати простой [5].

При виборі архітектури веб-додатку доцільно враховувати такі критерії:

– продуктивність: обсягом оброблюваних запитів та ефективністю кешування вимірюється час відгуку системи, аналіз проводиться за допомогою навантажувального тестування та порівняння затримок між різними спробами [4];

– відмовостійкість: здатність системи функціонувати далі при виході з ладу одного або декількох компонентів оцінюється через тестування відмовостійкості, моделювання відмов та резервування ресурсів [3];

– масштабованість: проводиться аналіз поведінки системи при збільшенні користувачів або запитів, система повинна ефективно обробляти зростаюче навантаження та забезпечувати балансуювання ресурсів [5];

– вартість впровадження та підтримки: проводиться економічний аналіз загальної вартості володіння, де враховуються початкові витрати, витрати на підтримку та масштабування системи [2].

Таким чином, враховуючи вищеперераховані критерії, методи оцінювання включають в себе проведення симуляцій навантаження, аналіз реальних кейсів та оцінку відмовостійкості за допомогою Chaos Engineering, що дозволяє виявити найбільш оптимальну архітектуру для веб-застосунку.

На практиці архітектура обирається залежно від конкретних вимог бізнесу та технічних умов. Для веб-застосунків із високим навантаженням найкраще підходять мікросервісна або безсерверна архітектури, що дозволяють гнучке масштабування та забезпечують стійкість до відмов. У випадках обмеженого бюджету доцільно починати з монолітної архітектури, поступово переходячи до мікросервісного підходу. Для критичних до доступності сервісів рекомендується використовувати мікросервісну архітектуру із балансуванням навантаження та резервними копіями. Якщо швидкість розробки є пріоритетною, варто звернути увагу на безсерверну архітектуру, яка мінімізує операційні витрати та автоматизує процес масштабування. Загалом, вибір архітектури має базуватися на ретельному аналізі вимог до доступності, продуктивності та економічної доцільності [1].

Подальші дослідження можна зосередити на аналізі впливу конкретних технологій, таких як Docker та Kubernetes, на забезпечення високої доступності. Також перспективним є вивчення використання хмарних платформ (AWS, Google Cloud, Azure) та їхніх вбудованих інструментів для підвищення відмовостійкості. Аналіз ефективності стратегій балансування навантаження та резервного копіювання також може надати корисні інсайти для майбутніх розробок.

Список використаних джерел:

1. Склярєнко О., Савченко Я., Литвиненко Л., Сушинський О. (2024). Архітектурні підходи до розробки масштабованих веб-застосунків. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 4(24), 341–350. <https://doi.org/10.28925/2663-4023.2024.24.341350> (дата звернення 11.02.2025).
2. What is monolithic architecture in software, Режим доступу: <https://www.techtarget.com/whatis/definition/monolithic-architecture> (дата звернення 11.02.2025).
3. What Are Microservices & How Do They Enhance Web Applications, Режим доступу: <https://www.gravitee.io/blog/what-are-microservices-how-do-they-enhance-web-applications> (дата звернення: 31.01.2025).
4. Serverless web applications: What you need to know before building one, Режим доступу: <https://newrelic.com/blog/best-practices/serverless-web-application> (дата звернення: 28.01.2025).
5. Sitnikov D., Ryabov O., Mishcheriakov I., Kovalenko A. A. Rough set based algebraic approach to modelling complex systems // The International Journal of Design & Nature and Ecodynamics. Vol. 13, No. 3, 2018. Pp. 324–329.