

УДК 004.273

АНАЛІЗ АРХІТЕКТУРНИХ ПАТЕРНІВ ДЛЯ ВИСОКОНАВАНТАЖЕНИХ ВЕБ-СИСТЕМ

Халтурін А.Д.

Харківський національний університет радіоелектроніки, Харків

Зростання вимог до продуктивності, надійності та здатності систем витримувати високі навантаження робить вибір архітектури критичним завданням. У роботі розглянуто чотири фундаментальні архітектурні патерни, що представляють різні філософії декомпозиції системи: монолітний, мікросервісний, безсерверний та подіє-орієнтований. У роботі розглянуто чотири фундаментальні архітектурні патерни: монолітний, мікросервісний, безсерверний та подіє-орієнтований.

Монолітна архітектура характеризується єдиною, нероздільною кодовою базою, де всі компоненти системи тісно інтегровані. Хоча цей підхід забезпечує простоту початкової розробки, розгортання та налагодження завдяки централізованій структурі, він демонструє значні обмеження при горизонтальному масштабуванні. Масштабування можливе лише шляхом дублювання всього застосунку на додаткових серверах, що призводить до неефективного використання ресурсів, особливо коли навантаження нерівномірно розподілене між компонентами. Крім того, монолітна архітектура ускладнює впровадження змін через високий ступінь взаємозалежності модулів та вимагає повного перезапуску системи навіть при оновленні окремих частин.

Мікросервісний підхід вирішує ці проблеми шляхом декомпозиції системи на набір незалежних, дрібнозернистих сервісів, кожен з яких відповідає за окрему бізнес-функцію. Це дозволяє здійснювати гранульоване масштабування окремих компонентів залежно від їхнього навантаження, покращує відмовостійкість через ізоляцію помилок та забезпечує технологічну гетерогенність — кожен сервіс може бути реалізований оптимальними для нього технологіями. Водночас цей патерн вносить суттєву

операційну складність, пов'язану з управлінням розподіленою системою, моніторингом багатьох незалежних процесів, мережевими затримками при міжсервісній комунікації та забезпеченням консистентності даних у розподіленому середовищі. Особливої уваги вимагають питання *service discovery*, розподіленого трейсингу та управління транзакціями.

Безсерверна (Serverless) архітектура пропонує пропонує модель FaaS (Function as a Service), де розробник повністю абстрагується від управління інфраструктурою, а хмарна платформа забезпечує автоматичне масштабування "з нуля" в залежності від реального навантаження. Модель оплати за фактичне використання робить цей підхід економічно ефективним для систем з непередбачуваними або періодичними навантаженнями. Однак безсерверна архітектура має характерні недоліки: "холодні старти" (затримки при ініціалізації неактивних функцій), обмеження часу виконання окремих функцій, ускладнене локальне тестування та значна прив'язка до постачальника хмарних послуг, що ускладнює міграцію між платформами.

Подіє-орієнтована архітектура забезпечує забезпечує максимальний рівень слабкого зв'язування компонентів через асинхронну комунікацію на основі подій, що підвищує стійкість та гнучкість системи. Компоненти взаємодіють не через прямі виклики, а через публікацію та обробку подій, що дозволяє легко додавати нові обробники без модифікації існуючих компонентів. Проте це ускладнює налагодження через відсутність явного потоку виконання, вимагає управління "узгодженістю в кінцевому підсумку" (eventual consistency) та часто потребує впровадження складних патернів, як-от CQRS (Command Query Responsibility Segregation) — розділення команд та запитів для оптимізації операцій читання і запису.

На рис. 1 схематично зображено чотири розглянуті архітектурні патерни. Рисунок візуалізує їхні фундаментальні структурні відмінності: від єдиного блоку «Моноліту» до розподіленої мережі, керованих функцій (FaaS) та асинхронної моделі обробки подій..

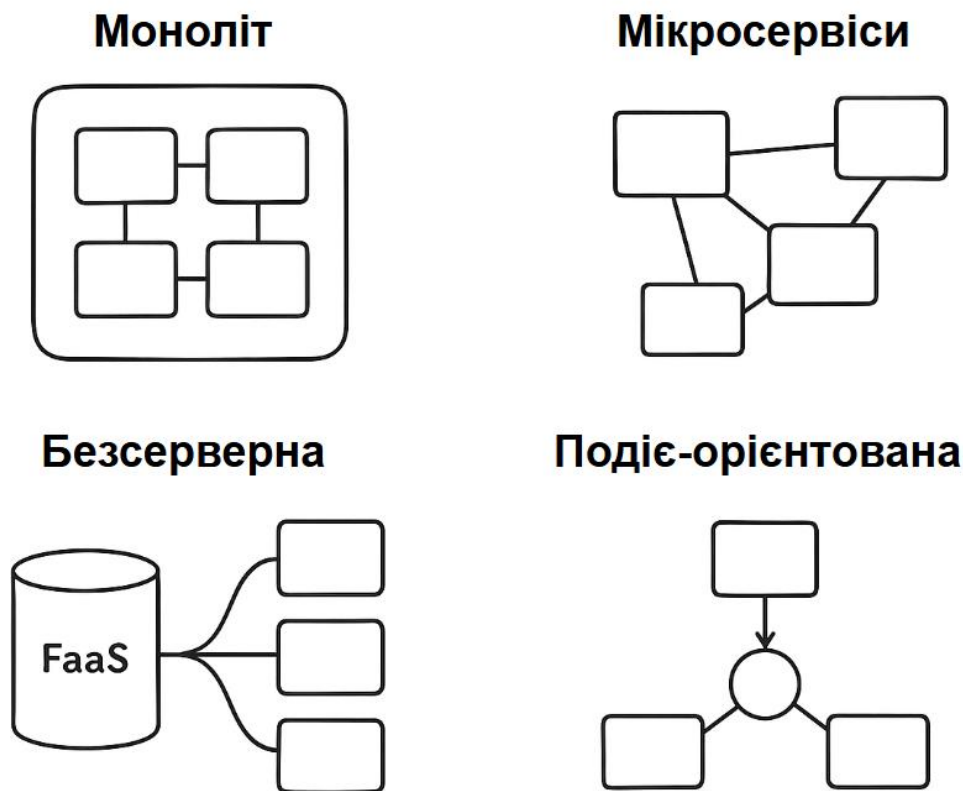


Рисунок 1 – Архітектурні патерни

Не існує єдиного "найкращого" рішення; вибір архітектури завжди є компромісом, що залежить від бізнес-вимог, очікуваного навантаження та можливостей команди. Застосування методів системного аналізу дозволяє зробити цей вибір обґрунтованим та стратегічно правильним.

Література:

1. M. L. Abbott and M. T. Fisher, The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. New York: Addison-Wesley, 2015.
2. S. Newman, Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2021.
3. "Serverless Architectures with AWS Lambda". 2018. [Он-лайн]. Доступно: <https://d1.awsstatic.com/whitepapers/serverless-architectures-with-aws-lambda.pdf>