

BÁO CÁO: So sánh Microservices và Monolithic Architecture trong phát triển hệ thống web hiện đại

I. Giới thiệu

Trong bối cảnh các hệ thống phần mềm ngày càng lớn và yêu cầu mở rộng cao, việc lựa chọn kiến trúc hệ thống trở thành quyết định chiến lược. Hai mô hình phổ biến là Monolithic Architecture và Microservices Architecture, đại diện cho hai triết lý thiết kế hoàn toàn khác nhau: tập trung và phân tán.

Báo cáo này không chỉ dừng ở việc so sánh, mà còn:

- Phân tích trade-offs kỹ thuật thực tế.
- Đánh giá tính phù hợp theo ngữ cảnh hệ thống.
- Thăm định độ tin cậy học thuật của các nguồn.

II. Phương pháp tìm kiếm thông tin

1. Chiến lược tìm kiếm

- Sử dụng Boolean Search:

- “microservices AND scalability AND performance”
- “monolithic architecture AND limitations AND case study”

- Lọc theo:

- Năm \geq 2015
- Có \geq 50 citations (Google Scholar)

2. Phân loại nguồn

- Paper (IEEE, Springer): Cơ sở học thuật
- Sách: Kiến thức hệ thống
- Industry Docs (AWS, Microsoft): Thực tiễn
- Expert Blog: Insight chiến lược

III. Phân tích chuyên sâu

1. Monolithic Architecture – Phân tích hệ thống

- Strong consistency và low latency.
- Gọi hàm nội bộ (in-process), không có network overhead.
- Transaction dễ quản lý (ACID).

Nhược điểm:

- Tight coupling \rightarrow exponential complexity growth.
- Scaling chủ yếu theo vertical scaling.
- Deployment rủi ro cao (single point of failure).

2. Microservices – Phân tích hệ phân tán

- Là một dạng distributed system.
- Đối mặt với network latency, partial failure và eventual consistency.

Ưu điểm:

- Independent deployment.
- Horizontal scaling.
- Fault isolation.

Nhược điểm:

- Distributed transaction phức tạp.
- Debugging khó.
- DevOps overhead lớn.

3. Trade-offs thực tế

Latency: Monolith thấp, Microservices cao hơn.

Consistency: Monolith strong consistency, Microservices eventual consistency.

Fault Isolation: Monolith kém, Microservices tốt.

Complexity: Monolith thấp ban đầu, Microservices cao ngay từ đầu.

Scaling Cost: Monolith cao, Microservices tối ưu hơn.

4. Case Study

Netflix:

- Chuyển từ Monolith sang Microservices.
- Tăng khả năng chịu tải nhưng tăng độ phức tạp vận hành.

Amazon:

- Áp dụng “2-pizza team rule”.
- Microservices giúp các team hoạt động độc lập.

Insight:

Microservices không tốt hơn một cách tuyệt đối; chúng trở nên cần thiết khi hệ thống đạt quy mô lớn.

IV. Bảng đánh giá nguồn

1. Dragoni et al. – Survey toàn diện – Độ tin cậy: ★★★★★
2. Newman – Thực tế nhưng có bias – Độ tin cậy: ★★★★★
3. Balalaie – Có dữ liệu thực nghiệm – Độ tin cậy: ★★★★★
4. Pahl – Thiên về lý thuyết – Độ tin cậy: ★★★
5. Taibi – Tổng hợp tốt – Độ tin cậy: ★★★
6. Kleppmann – Phân tích sâu distributed systems – Độ tin cậy: ★★★★★
7. Fowler – Uy tín cá nhân cao – Độ tin cậy: ★★★★★

8. AWS – Có bias sản phẩm – Độ tin cậy: ★★★
9. Microsoft – Mang tính hướng dẫn – Độ tin cậy: ★★★
10. Google Cloud – Tổng quan, ít chiều sâu – Độ tin cậy: ★★★

V. Đánh giá độ tin cậy

1. Bias phân tích

- AWS, Microsoft có xu hướng quảng bá microservices.
- Newman chịu ảnh hưởng từ kinh nghiệm cá nhân.

2. Academic vs Industry

Academic:

- Chính xác hơn.
- Ít bias.
- Cập nhật chậm.

Industry:

- Thực tiễn hơn.
- Cập nhật nhanh.
- Có thể mang tính thương mại.

Kết luận: Cần kết hợp cả hai nhóm nguồn để có góc nhìn cân bằng.

VI. Kết luận

Sai lầm phổ biến: “Microservices luôn tốt hơn”.

Thực tế:

- Microservices = Optimization for Scale.
- Monolith = Optimization for Simplicity.

Nguyên tắc lựa chọn:

- Startup nhỏ → Monolith.
- Hệ thống cần scale lớn → Microservices.
- Team kỹ thuật yếu → Monolith.
- Team mạnh về DevOps → Microservices.

VII. Tài liệu tham khảo

Dragoni, N. et al. (2017) Microservices: Yesterday, Today, and Tomorrow. Springer.

Newman, S. (2015) Building Microservices. O’Reilly Media.

Balalaie, A., Heydarnoori, A. and Jamshidi, P. (2016) ‘Microservices Architecture Enables DevOps’, IEEE Software.

Pahl, C. (2015) 'Containerization and the PaaS Cloud', IEEE.

Taibi, D. et al. (2017) 'Architectural Patterns for Microservices', Journal of Systems and Software.

Kleppmann, M. (2017) Designing Data-Intensive Applications. O'Reilly Media.

Fowler, M. (2014) Microservices. Available at: <https://martinfowler.com> (Accessed: 18 March 2026).

Amazon Web Services (2023) Microservices on AWS.

Microsoft (2022) Microservices Architecture Guide.

Google Cloud (2023) Microservices Overview.