My Stance

◀ On the microservices/microkernels debate

Display replies in nested form

Settings ⌄

My Stance

by Oi Lam Siu - Saturday, 25 January 2025, 4:20 AM

I would agree with the statement that microservices and microkernels represent the future of software architecture. Over the intervening decades, microservices and microkernel designs have indeed gained traction, especially in enterprise environments, where scaling and modular maintenance have become paramount.

On the one hand, microservices can offer a number of advantages: development teams can work on loosely coupled services, iterate quickly, and deploy updates without extensive impact on the entire system. As Fritzsch et al. (2019) suggest in their classification of refactoring approaches, organisations often decompose large monoliths into microservices precisely because of benefits like independent scalability, reduced downtime, and easier adoption of new technologies in specific components. These refactoring benefits mirror some of the same motivations behind microkernel design: isolating services (or OS components) so that faults in one area need not compromise the stability of the rest of the system.

On the other hand, a monolithic approach (be it an application or an operating system kernel) still has performance upsides in certain contexts, often because fewer context switches and fewer network calls can lead to better raw speeds. Even Andrew Tanenbaum's argument, as seen in the Tanenbaum–Torvalds debate, acknowledged the theoretical elegance of microkernels but accepted that performance overhead was a legitimate concern in practical systems of that era. Over time, improved hardware and more refined OS research (e.g. Mach, Chorus) have alleviated many of these overheads, which supports the case that microkernels and microservices are more appealing today.

Therefore, while it might be tempting to say that the success of Linux proves Torvalds was "right" about monolithic kernels, the landscape of commercial software has indeed shifted towards distributed, service-oriented styles reminiscent of microkernel philosophy. Still, the popularity of Linux and continued use of a monolithic kernel approach also prove that no single model is optimally correct for every use case. Ultimately, microservices and microkernels align well with modern development goals—especially continuous delivery, rapid scalability, and clearer fault isolation—and hence can be considered key pillars in contemporary and future architectures.

Reference

DiBona, C. & Ockman, S. (1999) Open Sources. 1st ed. Sebastapol: O'Reilly Media, Inc.

Fritzsch J., Bogner J., Zimmermann A. & Wagner S. (2019) From Monolith to Microservices: A Classification of Refactoring Approaches. In: Bruel JM., Mazzara M., Meyer B. (eds) Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. DEVOPS 2018. Lecture Notes in Computer Science, vol 11350. Springer.

Permalink Edit Delete Reply

◀ On the microservices/microkernels debate

Chat to us!