ABSTRACT

The Online Thesis Defense Application of the Faculty of Industrial Engineering (SOFI) has migrated from a monolithic to a microservices architecture to restore the system's declining performance and scalability after a sharp increase in users. The next challenge is selecting the most efficient communication protocol. This study quantifies REST's limitations in SOFI's thesis-registration module, evaluates the extent to which GraphQL mitigates those limitations, and formulates strategic guidelines for API selection in similar microservice ecosystems. Using an empirical software-engineering framework, we conducted a controlled experiment: two functionally identical services: REST and GraphQL were implemented in Go, containerized with Docker, and load-tested with Apache JMeter at 10, 50, and 100 virtual users. Performance was measured across five key parameters: response time, throughput, error rate, CPU usage, and memory usage. Results reveal a clear trade-off between latency and resource efficiency. Both architectures delivered comparable processing capacity and reliability (0.00% error rate) in all scenarios. REST achieved the lowest latency for simple single-item requests, with a minimum response time of 6 ms. In contrast, GraphQL demonstrated markedly superior resource efficiency: CPU utilization remained below 0.4%, and memory consumption averaged about 265 MB nearly half of REST's 452 MB. Consequently, REST remains ideal for lightweight services where minimal latency is critical, whereas GraphQL is preferable for modules that aggregate complex data or operate on constrained resources.

Keywords: REST API; GraphQL; Microservices Architecture; Performance Evaluation; Resource Efficiency; Response Time.