SECTION 3. DIGITAL PLATFORMS, SECURITY, AND ICT INFRASTRUCTURE

UDC 004.05

PERFORMANCE METRICS-BASED EVALUATION OF TOOLS FOR API CRUD LOAD TESTING

M. Mendielieva[1][0009-0002-4282-3147], Ph.D., S. Khlamov[2] [0000-0001-9434-1081], Ph.D., O. Vovk[3] [0000-0001-9072-1634], T. Trunova[4] [0000-0003-2689-2679], S. Lytvynenko[5] [0009-0003-2632-9082]

[1,2,3,4,5] Kharkiv National University of Radio Electronics, Ukraine
EMAIL: [1] mariia.mendielieva@nure.ua, [2] (corresp.) sergii.khlamov@gmail.com, [3] oleksandr.vovk@nure.ua, [4] tetiana.trunova@nure.ua, [5] serhii.lytvynenko@nure.ua

ОЦІНКА ІНСТРУМЕНТІВ ДЛЯ ТЕСТУВАННЯ НАВАНТАЖЕННЯ CRUD ЗА СТАНДАРТАМИ API НА ОСНОВІ ПОКАЗНИКІВ ПРОДУКТИВНОСТІ

М. Мендєлєва, Ph.D., С. Хламов, Ph.D., О. Вовк, Т. Трунова, С. Литвиненко

Харківський національний університет радіоелектроніки, Україна

Анотація. *У роботі порівнюється продуктивність k6, JMeter та Postman для навантажувального тестування REST API за 4 різних профілів робочого навантаження. Для оцінки було використано локально розгорнутий застосунок Node.js, а час відгуку, споживання CPU та RAM було проаналізовано за допомогою ANOVA. Результати демонструють значний вплив інструменту та робочого навантаження на результати перформанс тестування. JMeter забезпечив найкращий час відгуку взагалом, k6 показав більшу ефективність використання ресурсів, а Postman може добре підійти для швидкого та легкого тестування продуктивності API.*

Ключові слова: *тестування продуктивності, тестування навантаження, k6, Postman, JMeter, час відповіді, робоче навантаження.*

Abstract. *This study compares k6, Apache JMeter, and Postman for REST API load testing under 4 different workload profiles. A locally deployed Node.js application was used, and response time, CPU, and RAM consumption were analyzed using two-way ANOVA. The findings demonstrate significant effects of tool and workload on performance test results. JMeter provided the best response time,*

k6 showed greater resource efficiency, and Postman can be suitable for lightweight API performance testing.

Keywords: *performance testing, load testing, k6, Postman, JMeter, response time, workload.*

API performance is essential for reliable and scalable distributed systems [1], but evaluation results may vary depending on the testing tool and its execution mechanisms [2]. This is especially important for on-premise systems handling sensitive data, where realistic and efficient testing, together with adequate documentation and tool integration, is required, especially in conditions of remote and hybrid teamwork. This study compares k6, Apache JMeter, and Postman for REST API focusing on the effects of tool selection and workload profiles on overall and endpoint-level performance. k6 provides efficient JavaScript-based execution, JMeter supports diverse protocols and workloads with higher resource requirements, while Postman provides reusable API collections and predefined load profiles with more limited workload configuration.

A unified REST API workload was evaluated using k6, JMeter, and Postman under four load profiles (Ramp, Spike, Peak, and Fixed), with 10 independent runs per Tool × Load combination. Response time, CPU RAM consumption, workload, and error metrics were collected.

Two-way ANOVA assessed the effects of tool, load profile, and their interaction on performance metrics. Experiments on a locally deployed Node.js application with an embedded Better-SQLite3 database showed that both Tool and Load significantly affected average and p95 response time ($p < 0.001$, partial $\eta^2 > 0.6$). JMeter achieved the best response-time performance across most workloads, whereas k6 and Postman showed higher latency under high-load conditions (see Figure 1).

Endpoint-level analysis showed that performance varies by request type and workload, with JMeter showing the most stable performance for both read and write operations. The findings demonstrate that load testing tool selection significantly affects both performance and resource consumption. While JMeter provided the best response-time results and k6 showed better CPU efficiency but generally exhibited significantly higher response times for GET requests, and Postman required more resources overall, no tool was optimal across all workloads. Thus, load testing tool selection should

account for response time, resource consumption, and workload characteristics.

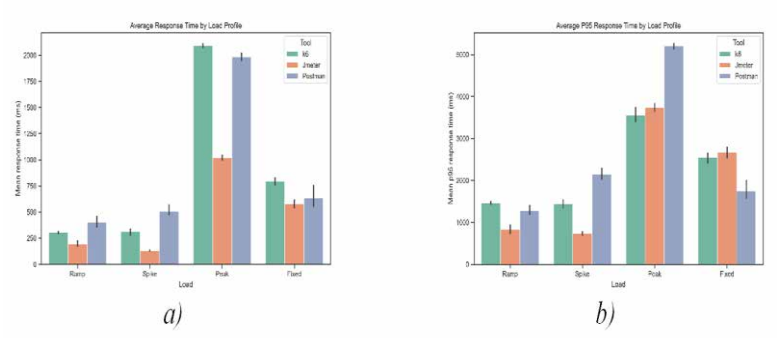

Figure 1. Mean response time: a) Average; b) p95 response time.

Figure 2 and resource consumption analysis showed that k6 showed advantages in CPU efficiency with higher memory consumption. Postman tool exhibited higher resource consumption across workload profiles.

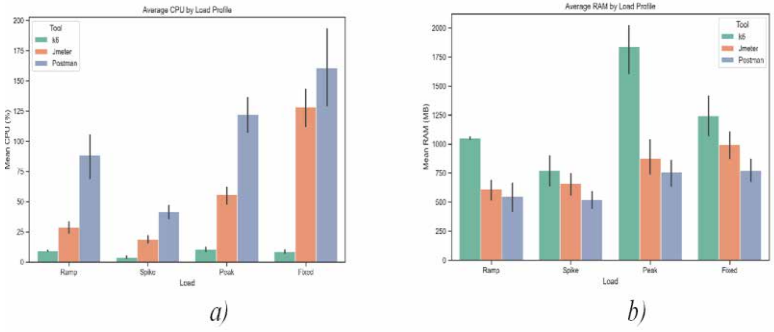

Figure 2. Average resources consumption by Load profile: a) CPU; b) RAM.

References

1. N. Li, J. Wang, C. Chen, H. Hu, Application of API automation testing based on microservice mode in industry software. *Proceedings of the International Conference on Algorithms, Software Engineering, and Network Security*, pp. 460-464, 2024.