UDC 004.42:502-048.34

A TOOL FOR MONITORING RESOURCE USAGE AND ENERGY CONSUMPTION OF SOFTWARE APPLICATIONS TO SUPPORT GREEN OPTIMIZATION

Laptinsky V.I.; Volkolovsky I.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Likhtarovitch I. I. – Senior Lecturer at the department of Foreign Languages

Annotation. This paper presents GreenDev Monitor, a tool for visualizing resource usage and energy efficiency of software applications. The system enables developers to evaluate the impact of code optimizations on energy consumption in real time and analyse performance changes during experimental testing. Experimental evaluation shows that green optimization reduces energy consumption by 33.9% while improving performance by 13.4%.

Keywords: software energy consumption, green technologies, resource monitoring, algorithm optimization, Spring Boot, energy efficiency

***Introduction*.** The problem of software energy efficiency is becoming increasingly relevant as computational workloads and cloud infrastructure usage continue to grow [1]. Research shows that non-optimized application code may consume 20–40% more energy than optimized equivalents [2, 3]. The concept of Green Software Engineering focuses on tools and methodologies that allow developers to assess and reduce the energy footprint of their applications [4]. At the same time, the lack of simple and accessible tools for measuring the energy impact of software decisions still limits the practical adoption of green optimization principles in everyday software development.

Existing energy profilers (Intel RAPL, ALEA, Green Metrics Tool) are primarily research-oriented and often require specialized hardware or complex setup procedures [5]. Therefore, creating a simple and accessible tool for everyday use by software engineers remains an important research and engineering task.

The objective of this work is to develop a web-based tool for monitoring resource usage and estimating energy efficiency, allowing developers to analyse and visualize the impact of optimizations on energy consumption and performance.

Main part. The developed GreenDev Monitor tool provides the collection and visualization of the following metrics:

– CPU load (CPU Load, %);

– RAM usage (Memory Usage, MB);

– estimated energy consumption (Estimated Energy, Wh).

Energy consumption is estimated using a linear model (Formula 1):

$$E \approx k_{CPU} \cdot \overline{CPU} \cdot t + k_{RAM} \cdot \overline{RAM} \cdot t, \qquad (1)$$

where is the estimated energy consumption (Wh);

k_{CPU} and k_{RAM} are calibration coefficients (W/% and W/GB);

$\overline{\text{CPU}}$ and $\overline{\text{RAM}}$ are average CPU load and memory usage;

t is execution time (hours).

The system architecture includes a metrics collection module, which is a background Java service that launches the application under test and periodically polls operating system counters; an energy calculation module that applies the mathematical model to the collected time series; a Spring Boot-based REST API that provides endpoints for managing measurement sessions and retrieving data; a PostgreSQL database that stores sessions, metrics, and experiment results. Such modular architecture allows the monitoring system to be easily extended with additional metrics or integrated with external profiling tools. This flexibility makes the tool suitable for both experimental research

and practical software development tasks.The monitoring results are presented to users through a web interface implemented as a responsive dashboard containing charts and tables. The interface allows developers to observe resource usage dynamics during application execution and analyse the impact of code changes on system performance and estimated energy consumption. Examples of the monitoring interface are presented in Figures 1–3.

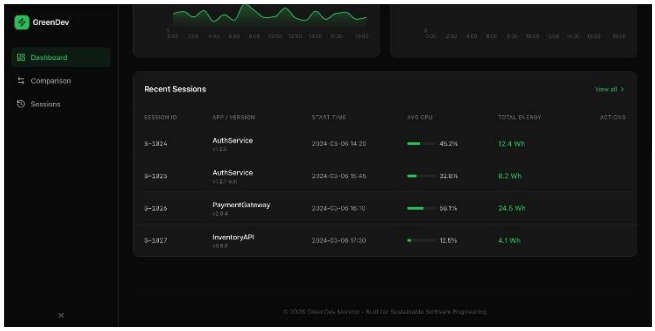

Figure 1 – Main monitoring dashboard with metric cards and system load charts

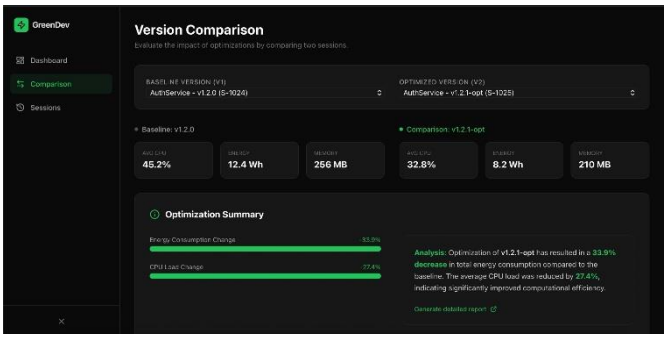

Figure 2 – Application version comparison page with energy consumption change visualization

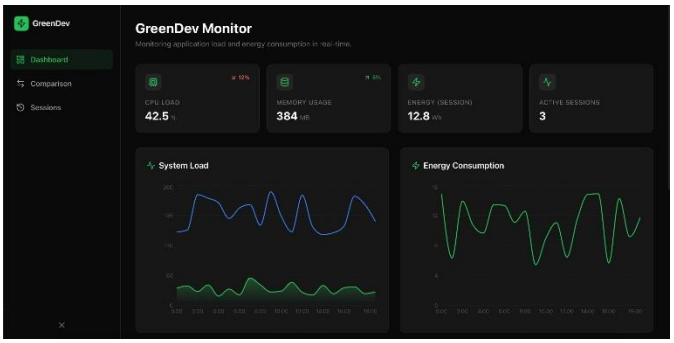

Figure 3 – Measurement sessions table with key metrics

The interface consists of three main pages: Dashboard (overview of metrics and real-time charts), Version Comparison (comparison of two versions of the application), and Session Details (detailed information about a specific session).

Table 1 – Technological stack of the prototype

Component	Technology
Backend	Java 17, Spring Boot 3.x
Database	PostgreSQL 15
Metrics collection	Java ProcessBuilder API
Frontend	HTML5, CSS3, JavaScript, Chart.js

The database includes three main tables: sessions (session metadata), metrics (detailed time series), and experiments (version comparison results). The metrics collection module launches the application under test as a child process and polls system counters (CPU, memory) every 1 second. Data are sent to PostgreSQL via REST API. After the session is finished, the energy calculation module aggregates metrics and computes estimated energy using the specified formula. The web interface is implemented in a dark color scheme with green accents in line with the Green IT concept. Charts are generated using the Chart.js library.

To evaluate the impact of optimizations on energy consumption and performance, an experiment was conducted using the AuthService Spring Boot web application. The methodology involved conducting a baseline measurement (v1.2.0), during which the application was subjected to load testing with 1000 sequential HTTP requests; identifying optimization points such as non-optimal SQL queries, redundant loops, and the absence of caching; implementing the optimizations in version v1.2.1-opt by adding database indexes, introducing Spring Cache, and replacing loops with Java parallel streams; performing a re-measurement of the optimized version; and finally carrying out a comparative analysis of the obtained results. The experiment results are presented in Table 2.

Table 2 – Comparison of metrics before and after optimization

Metric	Baseline	Optimized	Change (%)
Average CPU load, %	45.2	32.8	-27.4
Memory usage, MB	256	210	-18.0
Energy consumption, Wh	12.4	8.2	-33.9
Execution time, sec	187	162	-13.4

Optimization led to a 33.9% reduction in energy consumption with a simultaneous 13.4% performance improvement. The greatest effect was achieved through SQL query optimization (about 40% of the total energy savings), caching (30%), improved processing algorithms (20%), and elimination of redundant computations (10%). The results confirm that well-designed green optimization can simultaneously improve application performance and energy efficiency without significant trade-offs.

***Conclusion*.** The developed GreenDev Monitor tool provides developers with a simple and visual mechanism for assessing application energy efficiency. The experimental study demonstrated the practical value of the tool: green optimization reduced energy consumption by 33.9% while improving performance by 13.4%. Future work includes integration with hardware profilers (Intel RAPL), expanding support for various platforms, and implementing automatic optimization recommendations. This work contributes to the development of the Green Software Engineering concept and can serve as a foundation for further research in the field of energy-efficient programming. The proposed approach also demonstrates that integrating energy-awareness into everyday development tools can become an important step toward more sustainable and environmentally responsible software systems.

References

1. National Center for Biotechnology Information. Green IT Meaning in Energy Monitoring Practices (2023) [Electronic resource]. – Mode of access: https://pmc.ncbi.nlm.nih.gov/articles/PMC9875191/. – Date of access: 07.03.2026

2. IJFMR. Green Computing: Optimizing Algorithms for Energy Efficiency. International Journal For Multidisciplinary Research, 2025, Vol. 5 [Electronic resource]. – Mode of access: https://www.ijfmr.com/papers/2025/5/58349.pdf. – Date of access: 07.03.2026

3. Scientific Research Publishing. Green Software Engineering: A Study on Energy-Efficient Design (2025) [Electronic resource]. – Mode of access: https://www.scirp.org/journal/paperinformation?paperid=144329. – Date of access: 07.03.2026

4. University of Twente. A comprehensive monitoring toolkit for energy consumption measurement (2025) [Electronic resource]. – Mode of access: https://research.utwente.nl/en/publications/a-comprehensive-monitoring-toolkit-for-energy-consumption-measure/. – Date of access: 07.03.2026

5. Green-coding-solutions. Green Metrics Tool – Measure energy consumption (2022) [Electronic resource]. – Mode of access: https://github.com/green-coding-solutions/green-metrics-tool. – Date of access: 07.03.2026.