

# Architectural Tactics to Improve the Environmental Sustainability of Microservices: Evidence Briefing

**Author:** Xingwen Xiao < [x3.xiao@student.vu.nl](mailto:x3.xiao@student.vu.nl), [xingwen.xiao@student.uva.nl](mailto:xingwen.xiao@student.uva.nl) >

**Keywords:** Microservice, Green Software, Sustainability, Architectural Tactics, Energy

**Targeted Readers:** Software Engineers, Software Architects, DevOps

**What is a Rapid Review:** A concise and up-to-date review for industrial practitioners.

**What is not included in the briefing:** The exact implementation of the tactics, the statistics behind the research, and the proof of the effectiveness of these tactics.

Microservice sustainability is a field that may play an important role in the future with the growing concern for software sustainability. However, there is only a small amount of research on the architectural tactics to improve it.

## What Sustainability Aspects do Research Focus on

- **Energy Efficiency:** comparing effective output to energy input or energy used within a timeframe.
- **Carbon Efficiency:** comparing effective workload to carbon dioxide emissions.
- **Resource Efficiency:** comparing effective output to hardware (CPU, memory) utilization.

## Tactics Proposed to Improve Microservice Sustainability

- **Use elastic containers:** Traditional containers are pre-allocated fixed resources, hindering efficient resource utilization and complicating estimation. Elastic containers dynamically allocate resources based on monitoring each container's resource usage. This tactic improves resource efficiency. (e.g. Cusack, Greg, et al. "Efficient microservices with elastic containers." *Proceedings of the 15th International Conference on Emerging Networking EXperiments and Technologies*. 2019.)
- **Distribute pods to different nodes to leverage node properties:** Nodes vary in properties like their energy sources (solar, thermal). Assigning pods to different nodes helps to better utilize energy and improve load balancing. This tactic improves energy efficiency. (e.g. De Nardin, Igor Fontana, et al. "On revisiting energy and performance in microservices applications: A cloud elasticity-driven approach." *Parallel Computing* 108 (2021): 102858.)
- **Distribute microservices to different containers based on affinity:** Grouping similar microservices in the same container is advantageous when they have shared software dependencies or functions. It also helps in consolidating API requests for accessing similar resources within that container. This tactic improves resource efficiency. (e.g. F. H. L. Buzato and A. Goldman, "Optimizing Microservices Performance and Resource Utilization through Containerized Grouping: An Experimental Study," *2023 International Symposium on Computer Architecture and High Performance Computing Workshops*.)
- **Use predictive autoscaling algorithms:** Autoscaling dynamically adjusts resources for microservices based on demand fluctuations. It involves modifying pod replicas or hardware

resources. Advanced algorithms use predictions and new factors like carbon emissions to optimize allocation. This tactic improves energy efficiency and carbon efficiency. (e.g. Gebreweld, Haben Birhane. "Evaluating the energy consumption impact of a carbon aware autoscaling in microservice-based applications on the public cloud: a sustainability perspective." (2023).)

- **Use energy-efficient hardware:** Utilize new hardware like network interface cards with microprocessors to deploy microservices efficiently. This approach can reduce latency for API calls and save energy, especially for microservice tasks like handling API requests. This tactic improves energy efficiency. (e.g. Khairy, Mahmoud, et al. "SIMR: Single Instruction Multiple Request Processing for Energy-Efficient Data Center Microservices." *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2022.)
- **Use different proxies for different demands:** Microservices depend on API gateways, so choosing the right proxy is vital. Some proxies excel at high request volumes, while others prioritize energy efficiency during idle times. This tactic improves resource efficiency. (e.g. Nathaniel, Laurentius, et al. "Istio API gateway impact to reduce microservice latency and resource usage on kubernetes." *2023 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. IEEE, 2023.)

## Popular Infrastructures and Tools for Evaluating Microservice System Sustainability

- Cloud providers like Google Cloud, Microsoft Azure, and AWS.
- Kubernetes as the standard cluster configuration tool.
- Prometheus and Grafana for resource efficiency monitoring and visualization.
- Kepler for carbon emission and energy efficiency monitoring.

## Research Hotspot

- Applying predictive autoscaling algorithms, combined with machine learning and deep learning, to achieve better resource efficiency.
- Distributing pods to different nodes based on energy type or node load to improve energy efficiency.

## Challenges and Implications

- The ambiguity of the definition for carbon efficiency and energy efficiency models, along with the difficulty in measurement, hinders researchers from delving deep into the relevant tactics.
- Studies borrow concepts from other fields such as environmental sustainability, control theory, and machine learning. They contribute to research on architectural tactics for growth.
- Many research studies primarily focus on exploring application-level scenarios rather than delving into architectural innovation in the field of environmental sustainability.