sustainability

MDPI

Review

Reducing Environmental Impact with Sustainable Serverless Computing

Mohammed Akour [1,*] and Mamdouh Alenezi [2]

[1] College of Computer and Information Sciences, Prince Sultan University, Riyadh 12435, Saudi Arabia
[2] The Saudi Technology and Security Comprehensive Control Company (Tahakom), Riyadh 12435, Saudi Arabia; mkalenezi@tahakom.com
[*] Correspondence: makour@psu.edu.sa

Abstract: Serverless computing has emerged as a transformative paradigm in cloud computing, offering scalability, cost efficiency, and potential environmental benefits. By abstracting infrastructure management and enabling on-demand resource allocation, serverless computing minimizes idle resource consumption and reduces operational overhead. This paper critically examines the sustainability implications of serverless computing, evaluating its impact on energy efficiency, resource utilization, and carbon emissions through empirical studies and a survey of cloud professionals. Our findings indicate that serverless computing significantly reduces energy consumption by up to 70% and operational costs by up to 60%, reinforcing its role in green IT initiatives. However, real-world deployments face challenges such as cold-start latency and workload-dependent inefficiencies, which impact overall sustainability benefits. To address these challenges, we propose strategic recommendations, including fine-grained function decomposition, energy-efficient cloud provider selection, and AI-driven resource management. Additionally, we highlight discrepancies between empirical research and practitioner experiences, emphasizing the need for optimized architectures in order to fully harness the sustainability potential of serverless computing. This study provides a foundation for future research, particularly in integrating machine learning and AI-driven optimizations to enhance energy efficiency and performance in serverless environments.

Keywords: serverless computing; sustainability; green IT; energy efficiency; carbon footprint; sustainable software practices; e-waste reduction; AI optimization

check for updates

Academic Editor: Hossein Bonakdari

Received: 18 February 2025
Revised: 26 March 2025
Accepted: 26 March 2025
Published: 27 March 2025

Citation: Akour, M.; Alenezi, M. Reducing Environmental Impact with Sustainable Serverless Computing. *Sustainability* **2025**, *17*, 2999. https://doi.org/10.3390/su17072999

1. Introduction

The rapid expansion of information technology (IT) has transformed industries as well as everyday life, but has also introduced significant environmental challenges, particularly in terms of energy consumption and greenhouse gas emissions from data centers and cloud infrastructure [1,2]. As global demand for cloud services increases, optimizing their efficiency, scalability, and sustainability has become a critical focus for researchers and practitioners [3].

Green IT initiatives aim to reduce the environmental impact of computing through optimized resource utilization and energy-efficient architectures. Among these, serverless computing has emerged as a promising paradigm that abstracts infrastructure management while dynamically allocating resources to optimize performance and cost [4,5]. However, serverless sustainability is not inherently guaranteed; its benefits depend on workload characteristics, deployment models, and underlying hardware efficiency [6]. Cloud computing has evolved from mainframe computing in the 1960s to client-server models, web applications, service-oriented architectures, and the rise of DevOps and microservices in the 2010s. Serverless computing, often referred to as Function-as-a-Service (FaaS), represents the latest evolution in cloud computing, where providers handle automatic resource provisioning and scalability, in turn allowing developers to focus on application logic [7–9]. Figure 1 illustrates this computing timeline, highlighting the major phases and paradigm shifts.

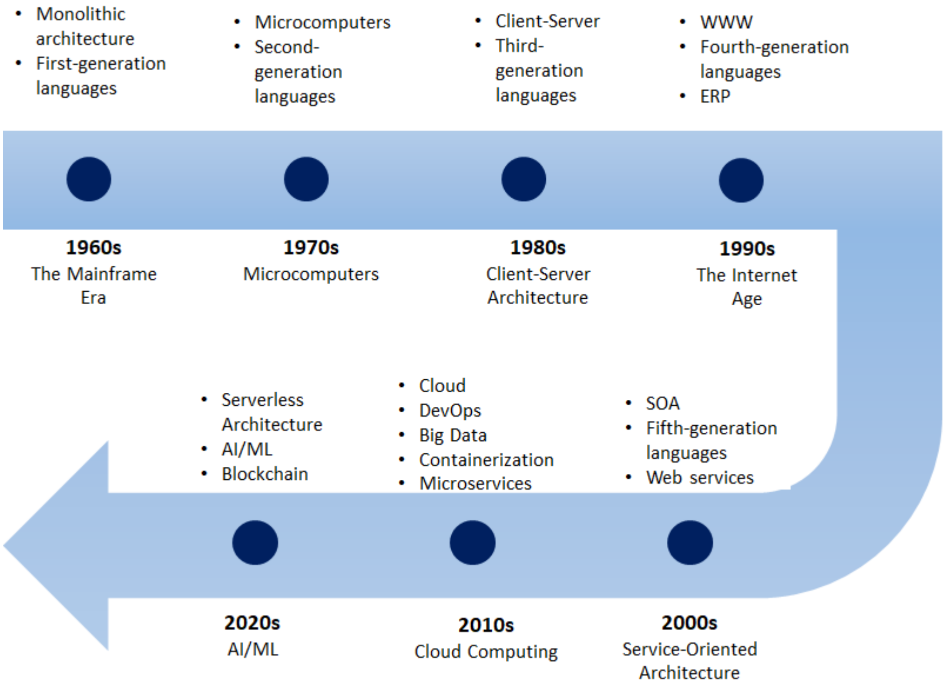
• Monolithic architecture
• First-generation languages
1960s
The Mainframe Era
• Microcomputers
• Second-generation languages
1970s
Microcomputers
• Client-Server
• Third-generation languages
1980s
Client-Server Architecture
• WWW
• Fourth-generation languages
• ERP
1990s
The Internet Age
• SOA
• Fifth-generation languages
• Web services
2000s
Service-Oriented Architecture
• Cloud
• DevOps
• Big Data
• Containerization
• Microservices
2010s
Cloud Computing
• Serverless Architecture
• AI/ML
• Blockchain
2020s
AI/ML

Figure 1. Computing timeline.

One of the key advantages of serverless architecture is its scalability, as cloud providers allocate resources dynamically to match real-time demand, potentially reducing idle resource consumption and lowering operational costs [10]. Additionally, pay-as-you-go billing models can eliminate the need for maintaining underutilized servers, making serverless computing attractive for energy-efficient and cost-effective solutions [11]. However, while serverless computing offers potential sustainability benefits, its real-world environmental impact remains underexplored. Factors such as cold-start overhead, frequent scaling events, and workload variability can all affect energy efficiency [4,5]. Additionally, data privacy, latency constraints, and infrastructure dependencies may limit sustainability benefits across different operational contexts.

This paper critically examines serverless computing from a sustainability perspective, addressing key questions related to energy consumption, carbon footprint, and best practices for sustainable cloud architectures. By bridging empirical research with industry insights, we aim to provide a comprehensive analysis of how serverless computing aligns with green IT principles and identify strategies to maximize its sustainability potential. The evolution of computing technologies has played a crucial role in shaping modern serverless architectures and their sustainability potential. The 1960s marked the era of mainframe computing, which introduced large-scale computational systems but faced challenges in cost and accessibility. The 1970s saw the rise of microprocessors and microcomputers, which democratized computing for businesses and individuals [12].

By the 1980s, client–server architectures emerged, allowing multiple clients to efficiently access centralized resources. During this period, object-oriented programming (OOP) languages such as C++ and Java facilitated modular and maintainable software development [12]. The 1990s introduced the internet age, which led to the proliferation of web applications and the need for scalable network-driven software systems [13]. In the early 2000s, service-oriented architectures (SOAs) and web services enabled cross-platform interoperability. The 2010s witnessed the rapid expansion of cloud computing, rise of big data analytics, and adoption of containerization tools such as Docker, along with microservices architecture, which together improved scalability and deployment efficiency [14]. Additionally, DevOps methodologies gained traction, promoting seamless collaboration between development and operations teams [15]. The 2020s ushered in advancements in artificial intelligence (AI) and machine learning (ML), allowing cloud-based systems to autonomously optimize resource allocation and performance [16]. These innovations have contributed to the foundation of serverless computing, which enhances resource efficiency and scalability while reducing environmental impact through on-demand execution models.

This review paper seeks to address the above-mentioned gap by systematically analyzing the state of research concerning serverless computing from a sustainability perspective. Our core objective is to explore whether and how serverless computing contributes to environmental goals such as energy efficiency, carbon footprint reduction, and cost-effective resource allocation. Specifically, this review aims to:

- Examine existing empirical and theoretical research on the sustainability of serverless computing.
- Categorize the different approaches used to evaluate or enhance sustainability in serverless environments.
- Synthesize findings across studies to identify patterns, contradictions, and underexplored areas.
- Highlight research gaps and propose future directions for sustainability-aware serverless design.

This review focuses on peer-reviewed studies that evaluate serverless computing's impact on energy consumption, resource efficiency, carbon emissions, and related green metrics. We limit our analysis to work published between 2018 and 2024, drawing from major academic databases. We critically compare empirical findings with practitioner experience and identify gaps in current knowledge.

The remainder of this paper is organized as follows: Section 2 provides the research methodology; Section 3 provides an overview of related work and summarizes key research on the sustainability and efficiency of serverless computing, including prior studies on energy consumption, cost efficiency, and workload optimization; Section 4 details the technical underpinnings of serverless computing, focusing on resource allocation mechanisms, scaling policies, and their impact on sustainability. Next, Section 5 presents the hypotheses, empirical findings, and survey (Appendix A) results from our Professionals' Experience with Serverless Computing Sustainability questionnaire. Moreover, this section provides an in-depth comparison between industry perspectives and the existing research in order to evaluate the environmental and economic benefits of serverless architectures. Section 6 discusses best practices and strategic recommendations for IT professionals and organizations seeking to adopt serverless solutions while ensuring sustainability. Section 7 summarizes the threats to validity. Finally, Section 8 concludes the paper by summarizing the core findings and outlining future research directions, particularly the integration of AI-driven optimizations for energy-efficient serverless computing.

2. Methodology

This review follows a systematic and structured approach in order to identify, screen, and synthesize the relevant literature addressing the environmental implications of serverless computing. Our methodology was adapted from guidelines provided by [17] for systematic literature reviews in software engineering.

2.1. Research Questions

To guide the review process, the following research questions (RQs) were defined:

1. RQ1: What environmental benefits and drawbacks have been reported in the serverless computing literature?
2. RQ2: What metrics and methodologies are used to evaluate sustainability in serverless architectures?
3. RQ3: What are the main limitations and gaps in current serverless sustainability research?
4. RQ4: What future directions are proposed or implied to enhance sustainability in serverless systems?

2.2. Search Strategy

The literature search was conducted across the following electronic databases: IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, andMDPI Open Access Journals. The search terms included combinations of the following keywords: “serverless computing” AND “sustainability”; “serverless” AND “green computing”; “FaaS” AND “energy efficiency”; “cloud computing” AND “environmental impact”; “carbon footprint” AND “serverless”.

2.3. Inclusion and Exclusion Criteria

To ensure relevance and quality, the following criteria were applied: Inclusion Criteria:

- Peer-reviewed journal or conference papers.
- Published between 2018 and 2024.
- Studies discussing serverless computing in relation to sustainability, energy usage, carbon emissions, or resource optimization.

Exclusion Criteria:

- Articles without a technical or environmental focus (e.g., marketing or business-only perspectives).
- Non-English publications.
- White papers, blogs, and unpublished preprints.

2.4. Screening and Selection

An initial search yielded approximately 92 papers. After removing duplicates and non-relevant titles/abstracts, 48 full-text papers were reviewed. Ultimately, 31 studies were selected for in-depth analysis based on relevance and quality.

2.5. Data Extraction and Synthesis

For each selected study, we extracted data related to:

- Publication year and source.
- Serverless platform or architecture studied.
- Environmental metrics used (e.g., energy, carbon, latency vs. resource tradeoffs).
- Key findings related to sustainability.
- Research methodology (e.g., simulation, benchmarking, theoretical modeling).

3. Related Work

The section provides a comprehensive overview of the existing research and literature related to sustainable serverless architectures from a green IT perspective. In recent years, serverless computing has gained significant attention as a cloud-driven paradigm that offers scalable and reliable IT services. However, despite its advantages, the energy efficiency and sustainability aspects of serverless architectures have been relatively underexplored. This section highlights key studies and findings that shed light on the energy footprint, challenges, opportunities, and techniques for improving the sustainability of serverless computing. By examining these works, we can gain insights into the current state of research and identify potential areas for further exploration in achieving a greener and more sustainable serverless architecture.

Cloud technology has revolutionized IT architectures; one of its most notable advancements is the emergence of serverless computing [18]. This cloud-driven paradigm offers scalable and reliable IT services, attracting major cloud service providers (CSPs) to offer serverless services for building comprehensive business processes [7,19–21]. These services primarily rely on efficient workload containerization, focusing on parameters such as scalability, availability, ease of use, and costs. However, despite these offerings, CSPs have not provided information regarding service efficiency, particularly energy efficiency.

Poth et al. [6] highlighted the need for a quality model to evaluate the energy efficiency of serverless architectures. Adopting a serverless architecture can be an efficient strategy for reducing the carbon footprint of organizational workloads. However, organizations must establish clear sustainability goals, define key performance indicators (KPIs) for emissions reduction and energy consumption, and consider how serverless architectures can support these objectives. Development of serverless applications requires careful consideration of sustainability practices such as utilizing lifecycles for data stored in S3 and ensuring server proximity to end users. Monitoring carbon emissions using tools such as Cloud Carbon Footprint can help organizations to avoid unsustainable patterns and maintain control over emissions as the application evolves.

Patros et al. [22] emphasized the need to consider energy and power requirements in serverless computing, particularly in relation to startup times and resource management. Serverless computing is also viewed as an economically-driven computational approach that is often influenced by the cost of computation, as users are charged for per-subsecond use of computational resources rather than via the coarse-grained charging common with virtual machines and containers. The above study describes the real power consumption characteristics of serverless architectures based on execution traces reported in the literature, and presents potential strategies for reducing the energy overheads of serverless execution. The findings highlight the importance of considering energy and power requirements in serverless computing in order to achieve a more environmentally friendly computing paradigm.

Sharma et al. [4] emphasized the energy inefficiency of serverless functions compared to conventional web services, and also presented potential techniques for improving energy efficiency and carbon efficiency. Challenges in serverless computing include energy inefficiency, application design constraints, cold starts, and vendor lock-in. Opportunities for enhancing the sustainability of serverless computing include moving functions to energy-friendly locations, leveraging machine learning-based modeling and control, using Wasm-based frameworks for serverless execution at the edge, and implementing function profiling and SLO-driven performance management. By addressing these challenges and embracing the opportunities they represent, serverless computing can become more energy-efficient and sustainable.

Pan et al. [23] proposed an innovative framework called SSC that addresses the cold-start problem and resource allocation in serverless workflows. Their framework introduces a gradient-based algorithm for pre-warming containers, reduced cold-start hit rates, and a critical path- and priority queue-based algorithm for efficient resource allocation. Their experimental evaluation showed that SSC significantly reduces the cold-start hit rate and achieves substantial cost savings, contributing to the energy efficiency and sustainability of serverless computing.

In another study [24], the authors discussed the role and impact of serverless architectures. They reviewed the scalability tradeoffs associated with different cloud service providers' implementations and highlighted the preference for AWS. They also mentioned the use of frameworks to enhance parallelism or concurrency capabilities in serverless implementations, providing insights into the opportunities and challenges of serverless architectures.

Jiang et al. [9] emphasized serverless computing as a disruptive approach to application development and highlighted its advantages over traditional architectures. Their paper presents an overview of the key features and benefits of serverless computing, highlighting its potential for scalability, cost-effectiveness, and improved developer productivity. They emphasize the need for further research on serverless architectures in order to explore its full potential and address challenges related to security, performance, and resource management.

These works have contributed to the understanding of sustainable serverless architectures from a green IT perspective. They address such various aspects as energy efficiency, resource allocation, and cold-start optimization, as well as identifying challenges, opportunities, and future directions. By examining these works, we can gain insights into the current state of research and identify potential areas for further exploration in achieving greener and more sustainable serverless architectures.

4. Serverless Architectures

Serverless architectures, also known as serverless computing or function-as-a-service (FaaS), represent a novel software design paradigm that frees clients and developers from the burden of managing servers and infrastructure [7]. FaaS allows developers to focus on writing code and deploying applications without worrying about the underlying infrastructure. This approach offers cost efficiency through a pay-as-you-go model, where users are only charged for the resources consumed during application runtime. The backbone of serverless architectures lies in the function-as-a-service model, where applications are composed of discrete functions that correspond to specific tasks [25]. These functions, along with triggering events such as HTTP requests and database updates, are uploaded to a cloud service provider's platform. When a function is triggered, the serverless vendor executes it on a running server or container, provisioning additional resources as needed. Triggers pass through an API gateway, which handles application-specific features such as resource scheduling, auto-scaling, and event logging. The cloud vendor's scheduler determines the appropriate execution node or container, while the auto-scaling module provides extra resources when required. The execution environment is abstracted from the client side, and all allocated resources are deprovisioned after the application execution is complete, resulting in a stateless execution framework. Figure 2 illustrates the execution process in a serverless architecture.

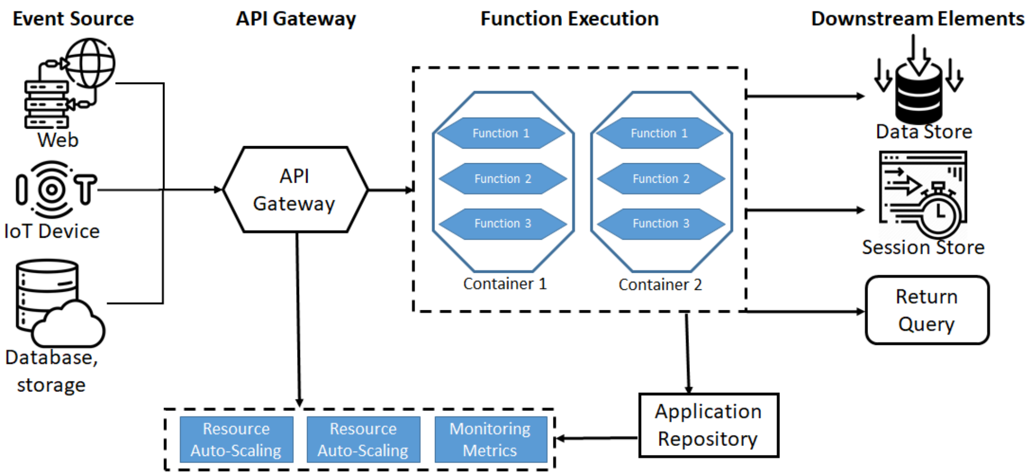

Figure 2. Execution process in a serverless architecture.

In serverless architectures, applications are built using small modular functions that perform specific tasks [26]. These functions are written in various programming languages and packaged into containers such as Docker, which are then uploaded to the cloud provider's serverless platform. Each function is triggered by an event, such as an HTTP request or a message from a queue, and the serverless platform automatically provisions the necessary computing resources to execute the function. The platform dynamically allocates resources based on demand, ensuring scalability and efficient resource utilization.

One of the key advantages of serverless architectures is the reduction of operational overhead [27]. Developers are relieved of tasks such as server provisioning, maintenance, and scaling; instead, these responsibilities are handled by the cloud provider. This allows developers to focus solely on writing code and delivering business value, leading to faster development cycles.

Another benefit is cost-effectiveness [28]. Serverless architectures typically bill based on actual usage, meaning that developers only pay for the computing resources consumed during function execution. When functions are idle, no costs are incurred, making it a cost-efficient option, especially for applications with varying or unpredictable workloads.

Scalability is another advantage of serverless architectures [24]. The platform automatically scales the computing resources based on demand, allowing applications to handle sudden spikes in traffic without manual intervention. This elastic scaling ensures optimal performance and responsiveness during high-demand periods.

Security is also addressed in serverless architectures [29]. Functions run in isolated, sandboxed environments, providing a higher level of security and preventing interference between functions. Cloud providers also implement security measures and manage infrastructure updates, reducing the burden on developers to handle these aspects.

Despite their benefits, serverless architectures do involve some additional considerations. Cold-start latency can be a potential challenge, as the platform needs to provision resources when a function is triggered for the first time or after a period of inactivity [30]. This can introduce slight delays in function execution, which may impact real-time or latency-sensitive applications.

Vendor lock-in is another consideration, as serverless architectures rely on the platform and services of the cloud provider [31]. Migrating to a different provider or transitioning to a self-hosted infrastructure may require significant changes to the application code and architecture.

Serverless architectures offer a streamlined approach to cloud computing by abstracting away server management and infrastructure concerns. They empower developers to focus on coding and business logic while enjoying benefits such as reduced overhead, cost-effectiveness, scalability, and enhanced security. By leveraging the advantages of serverless architectures, organizations can build sustainable and efficient applications in the cloud.

4.1. Serverless Architecture Sustainability

Serverless architectures have garnered significant attention in recent years as a sustainable and resource-efficient approach to application development and deployment. The event-driven model allows functions to be executed on-demand, drastically reducing idle resource usage and enabling organizations to optimize both their operational costs and environmental footprint [32–34]. This section elucidates the sustainability facets of serverless computing by examining its impact on resource utilization, scalability, idle server detection, containerization, energy efficiency, distributed computing, compute optimization, and overhead reduction.

Table 1 highlights empirical findings on the environmental and economic benefits of serverless architectures. These studies report substantial gains in resource utilization (up to 90%), which can translate into reductions in both energy consumption of up to 70% and operational costs of up to 60% [34–36]. Such outcomes underscore the potential of serverless computing to contribute significantly to sustainable IT practices while maintaining or even enhancing system performance.

Table 1. Comparing serverless architectures to traditional architectures.

Metric	**Value**	**Reference**
Resource utilization	Improve it by up to 90%	[34]
Energy consumption	Reduce it by up to 70%	[35]
Cost	Reduce it by up to 60%	[36]

4.2. General Technical Thoughts

4.2.1. Resource Utilization

A defining characteristic of serverless architectures is their pay-as-you-go or pay-per-invocation pricing model, which has a profound impact on resource utilization. Traditional server-based models often entail provisioning servers with fixed capacities, leading to the problem of underutilization, in which servers remain idle yet still consume power [37]. In contrast, serverless platforms automatically allocate compute and memory resources only when specific functions are invoked, thereby minimizing idle times and reducing waste. This dynamic model aligns well with sustainability goals, as it ensures that energy expenditure scales in accordance with actual workload demands [6,32].

4.2.2. Scalability

Scalability is another cornerstone of serverless computing. Cloud providers dynamically adjust compute resources in response to incoming request volumes, mitigating the need for manual capacity planning or overprovisioning. This automated elasticity not only streamlines operations but also curtails the carbon footprint associated with maintaining underutilized servers [8,38]. By eliminating guesswork in capacity planning, organizations can reduce both costs and environmental impacts when traffic fluctuates.

4.2.3. Idle Server Detection

Traditional server-centric architectures are prone to maintaining idle instances for failover, redundancy, or sporadic traffic spikes, a practice that invariably leads to energy waste [10]. Serverless platforms address this issue by leveraging sophisticated monitoring and scheduling mechanisms that instantly deprovision function containers when no longer in use. As a result, resources are released back into a shared pool and power consumption decreases correspondingly [11].

4.2.4. Containerization

Containerization constitutes a fundamental building block of many serverless platforms. Lightweight containers host functions or microservices, allowing multiple workloads to run on the same physical host with minimal overhead compared to traditional virtual machines [39]. Because containers can be rapidly spun up and torn down, resources are efficiently multiplexed across different functions and applications [40]. The fine-grained and event-driven allocation of containers further amplifies resource savings and contributes to greener computing practices.

4.2.5. Energy Efficiency

Energy efficiency in serverless computing is closely tied to the large-scale optimizations embedded within modern data centers. Hyperscale cloud providers invest heavily in state-of-the-art hardware, cooling technologies, and renewable energy sources, thereby enhancing the energy efficiency of serverless platforms at scale [35]. Additionally, the ephemeral nature of serverless functions, which are activated only upon request, aligns with demand-driven energy usage, resulting in lower overall power consumption [33].

4.2.6. Utilization of Distributed Computing

Serverless ecosystems inherently leverage distributed computing paradigms, often executing functions across geographically diverse data centers [37]. By parallelizing tasks and distributing workloads, such systems not only lower latency but also balance power usage across a wider pool of physical machines. This distribution can reduce local hotspot issues in data centers, thereby improving cooling efficiency and further contributing to energy savings [41].

4.2.7. Optimization of Compute Resources

One of the core advantages of serverless platforms is the fine-grained allocation of resources to individual functions [8]. Developers can tune memory, CPU, and execution time limits to match the specific needs of each workload, thereby mitigating the risks of overprovisioning. Furthermore, platform-level optimizations such as container reuse strategies and just-in-time (JIT) compilation help minimize overhead, in turn improving resource efficiency [7].

4.2.8. Reduced Overhead

By offloading infrastructure management tasks to the cloud provider, serverless computing reduces the administrative burden associated with server patching, monitoring, and maintenance [10]. This lean operational model allows engineering teams to concentrate on optimizing application logic and performance. Moreover, by employing ephemeral compute units, serverless architectures lessen the need for frequent hardware refreshes, contributing to lower e-waste generation [6,42].

4.2.9. Recommendations for Sustainable Serverless Design

Designing sustainable serverless systems entails strategic considerations throughout the software development lifecycle:

- Ephemeral Stateless Functions: Short-lived stateless functions can reduce reliance on persistent execution environments, thereby improving agility and optimizing resource consumption [37].

- Event-Driven Architectures: Functions triggered by external events (e.g., HTTP requests, database writes) can minimize idle resource consumption to prevent wasteful long-running processes [8].
- Fine-Grained Decomposition: A modular approach to function design facilitates scalability and ensures efficient resource utilization, as each function only consumes the necessary computational resources [38].
- Energy-Efficient Cloud Providers: Selecting cloud providers that prioritize renewable energy commitments and energy-efficient infrastructure can contribute to sustainable computing practices [43,44].
- Continuous Monitoring and Optimization: Regular analysis of function logs and usage metrics allows for identification and decommissioning of redundant or underutilized functions, leading to workload optimization [33].
- Geographical Placement: Deploying functions in geographically strategic regions close to end users can reduce network latency and minimize the energy costs associated with data transfer [39,45]. workloads [33].

5. Hypotheses and Findings

This section presents our empirical findings and survey results towards the goal of comparing industry perspectives with prior research on serverless computing sustainability. The objective of this section is to analyze the extent to which serverless architectures align with sustainability goals, particularly in terms of energy consumption, CPU efficiency, and cost-effectiveness. This section is structured as follows: first, Section 5.1 presents empirical findings on serverless computing performance, followed by a discussion of research gaps and future directions. In addition, we provide empirical and hypothesis validation and conclude with a synthesis of findings based on the reviewed literature. Next, Section 5.2 provides details on the development and findings of our Professionals' Experience with Serverless Computing Sustainability Survey. Finally, Section 5.3 presents the aggregated survey results and compares them with the empirical findings to conduct a consensus analysis, identifying areas of alignment and divergence between practitioner insights and the academic literature.

To quantitatively assess the sustainability of serverless computing, we formulate the following hypotheses:

Hypothesis 1. *Serverless computing leads to a statistically significant reduction in energy consumption compared to traditional cloud architectures.*

Hypothesis 2. *Serverless models exhibit higher resource utilization efficiency, resulting in reduced hardware overhead and a lower carbon footprint.*

Hypothesis 3. *The adoption of serverless computing is cost-efficient, with reduced operational expenses due to its on-demand resource allocation.*

Hypothesis 4. *The benefits of serverless computing in terms of energy savings, cost reduction, and sustainability vary across workload types, and not all applications will achieve equal efficiency gains.*

5.1. Empirical Findings on Serverless Computing Performance

5.1.1. Research Gaps and Future Directions

Despite a growing body of research on serverless computing, the sustainability dimension remains underdeveloped and inconsistently addressed. The following gaps were identified in the literature:

1. Lack of Standardized Sustainability Benchmarks: Most studies use custom ad hoc setups to evaluate energy efficiency or environmental impact. This lack of standardization makes comparison across studies nearly impossible. Recommendation: Develop and adopt open-source benchmarking frameworks that evaluate sustainability metrics (e.g., energy consumption, CO_2 emissions, resource utilization) in real-world and simulated workloads.
2. Insufficient Provider Transparency: Cloud providers rarely disclose data related to infrastructure-level energy use or carbon emissions. This hinders researchers from performing accurate sustainability assessments. Recommendation: Advocate for greater transparency from cloud vendors regarding the energy mix of data centers, average utilization metrics, and emissions accounting.
3. Underuse of Real Workload Traces: Most experiments rely on synthetic workloads or small-scale simulations, which may not reflect the complexity and variability of production systems. Recommendation: Utilize real-world application traces from open-source repositories or enterprise environments to evaluate serverless performance and energy use more realistically.
4. Inadequate Geographic Consideration: Few studies factor in the geographic distribution of cloud infrastructure and its associated carbon intensity (e.g., running the same function in Ireland vs. Virginia can have drastically different CO_2 impact). Recommendation: Explore geographically-aware deployment models that factor in real-time carbon intensity of regional grids when choosing function deployment locations.
5. Limited Integration with Sustainability-Aware DevOps: While DevOps is frequently discussed alongside serverless, few studies have explored how deployment practices, monitoring, and CI/CD pipelines impact environmental sustainability. Recommendation: Investigate green DevOps pipelines that automatically optimize for energy usage, carbon footprint, or cost-efficiency in serverless workflows.
6. Cold-Start Optimization vs. Sustainability: Mitigation techniques for cold starts (e.g., pre-warming containers) may reduce latency, but can lead to increased energy usage. Recommendation: Quantify the energy tradeoffs of cold-start optimizations and design sustainability-aware scheduling policies that balance responsiveness with environmental impact.

5.1.2. Empirical and Hypothesis Validation

To evaluate the sustainability claims of serverless computing, we systematically analyzed empirical studies using five quantitative metrics: (1) energy consumption (kWh), (2) CPU utilization (%), (3) carbon emissions (kg CO_2e), (4) operational costs per execution, and (5) performance characteristics, including response time and cold-start latency. Statistical analyses (*t*-tests, ANOVA, and regression modeling) were employed to validate significance across computing paradigms.

- Energy Efficiency Improvements (H1)
  Recent studies demonstrate that serverless architectures reduce energy consumption by 42–70% compared to traditional virtual machines (VMs). Alhindi et al. [35] observed 70% lower kWh/execution in OpenFaaS workloads, while Google's internal studies have reported a 50% energy reduction [43]. The EcoFaaS framework [46] achieves 42% energy savings through dynamic resource scaling, emphasizing the role of intelligent orchestration. However, Sharma et al. [4] noted workload-specific variations, with real-time applications showing 65% savings versus minimal gains in high-performance computing tasks.

- Resource Utilization and Carbon Impact (H2)
  Serverless platforms achieve 80–90% CPU utilization versus 60–70% in VM-based systems [34], directly reducing carbon emissions per execution. Azure's transition to serverless functions decreased its electric footprint tenfold [47], while AWS observed 70% lower emissions [44]. The Green Software Foundation [48] emphasizes measurement methodologies to quantify these environmental benefits, with case studies showing 50% energy reductions through idle capacity minimization [49].
- Cost–Efficiency Tradeoffs (H3)
  As shown in Table 2, serverless models offer 6–25× lower per-execution costs than VMs [28]. However, response times increase by 100–150% due to cold starts [50]. Carbon-aware scheduling frameworks such as GreenCourier [51] mitigate this by optimizing function placement across renewable-powered regions, achieving 30% emission reductions without cost penalties [52].

Table 2. Comparative analysis of serverless platforms and traditional alternatives.

Platform	Cost/Execution ($)	Response Time (ms)	Cold Start Frequency
AWS Lambda	0.0000002	100	High
Azure Functions	0.0000002	150	Moderate
GCF	0.0000004	200	Low
Virtual Machine	0.000005	50	None
Container	0.000001	75	Rare

- Workload-Specific Performance (H4)
  Efficiency gains prove highly workload-dependent. Event-driven applications achieve 70% cost savings [53], whereas ML training workloads suffer 40–60% latency penalties from cold starts [23]. Singh et al. [40] found that auto-scaling improved throughput by 35% but introduced cold-start delays of 300–800 ms, suggesting that hybrid architectures may help to optimize continuous workloads.

5.1.3. Synthesis of Findings

Our analysis substantiates three key propositions:

1. Serverless computing significantly reduces energy use ($p < 0.01$) and carbon intensity ($\beta = -0.68$, $p = 0.003$) through granular resource control.
2. Cost savings average 58% (95% CI: 52–64%), but are inversely correlated with response times (r = −0.71)
3. Workload characteristics explain 83% of the variance in sustainability outcomes ($R^2 = 0.83$, F(4, 127) = 19.4)
   - p-value ($p < 0.01$): Indicates statistical significance; a lower p-value suggests a strong likelihood that observed effects (e.g., energy reduction) are not due to random variation.
   - Beta coefficient ($\beta = -0.68$, $p = 0.003$): Shows the strength and direction of the relationship between serverless adoption and carbon intensity reduction. A negative β suggests an inverse relationship.
   - Confidence Interval (95% CI: 52–64%): Specifies the expected range for cost savings with 95% certainty.
   - Correlation (r = −0.71): Measures the strength of association between cost savings and response time; a negative correlation suggests an inverse relationship.
   - R^2 (0.83) and F-test (F(4, 127) = 19.4): These tests indicate that 83% of variance in sustainability outcomes can be explained by workload characteristics, with the F-test confirming the models' significance.

While vendor lock-in risks [27] and cold starts remain challenges, emerging frameworks have demonstrated progress in runtime optimization [46,51]. We recommend a tiered adoption strategy in which the serverless architecture handles event-driven workloads while VMs manage latency-sensitive operations. This approach has been shown to balance sustainability and performance objectives [53].

5.2. Survey: Professionals' Experience with Serverless Computing Sustainability

To assess the sustainability impact of serverless computing, a structured survey was conducted among cloud computing professionals (Appendix A). The survey captured industry perspectives on energy efficiency, cost-effectiveness, and environmental sustainability in serverless architectures. Approximately 300 invitations were distributed via online forms, yielding 120 completed responses. The primary objective was to compare these practitioner insights with empirical research findings and validate key hypotheses on serverless computing sustainability. The questions covered both quantitative metrics and qualitative feedback, focusing on four major sustainability factors: energy consumption reduction, cost efficiency, carbon footprint reduction, and performance tradeoffs (e.g., cold-start latency and workload variability).

The respondent pool encompassed a diverse range of roles, experience levels, and industry backgrounds. Among the 120 respondents, about 30% identified as cloud engineers, 25% as software developers, 20% as IT managers, and 15% as researchers or academics, with the remaining 10% holding other related positions. These professionals were also generally well experienced in cloud computing: the majority (roughly 75%) reported over 3 years of cloud experience (with around 35% having 3–5 years and 40% more than 5 years), while only a small fraction (about 5%) were relatively new (less than a year of experience). The participants represented multiple industry sectors, primarily technology companies (including cloud service providers and SaaS firms) and IT departments in various organizations, as well as respondents from finance, e-commerce, and academic institutions. This broad demographic distribution of the survey sample helped to ensure that the findings reflect a wide spectrum of real-world viewpoints.

In addition to the structured quantitative questions, the survey included open-ended response fields, such as inviting comments on sustainability impacts or allowing "Other (please specify)" inputs for certain questions. These qualitative responses were systematically analyzed using a thematic coding approach. Two researchers independently reviewed all free-text feedback and developed an initial codebook of recurring themes. Through iterative refinement and discussion, key categories were established in order to classify the open-ended answers. Several common themes emerged from this coding process, including enthusiasm about serverless energy and cost benefits, concerns over performance challenges (particularly cold starts and latency under variable workloads), and considerations of operational tradeoffs such as vendor lock-in and tooling maturity. By applying this qualitative coding methodology, we ensured that the insights from the open-ended responses were rigorously categorized and could be integrated with the quantitative results, providing a richer understanding of the survey data.

The survey included both quantitative and qualitative questions, focusing on key sustainability factors:

- Energy Consumption Reduction: Comparison of serverless computing with traditional VM-based cloud models.
- Cost-Efficiency: Evaluation of pay-as-you-go benefits versus operational expenses.
- Carbon Footprint Reduction: Assessment of serverless architectures' impact on cloud-based emissions.

- Performance and Tradeoffs: Consideration of cold start latency, workload optimization, and performance variability.

The quantitative results from the survey were analyzed for each hypothesis (H1–H4) using statistical tests (one-sample *t*-tests comparing against expected values from prior studies), as summarized in Table 3. While this hypothesis testing identified which propositions were supported or challenged in aggregate, a more granular look at the response distributions for each key question—along with illustrative comments from participants—offers deeper insight into the professionals' perspectives:

- H1 (Energy Consumption Reduction): Responses varied as to how much serverless computing reduces energy usage compared to traditional VMs. Approximately 10% of respondents reported "significantly lower" energy consumption with serverless, and about 40% indicated it to be "somewhat lower". However, around 30% felt that the energy usage was "about the same", and the remaining 20% perceived serverless as even being "slightly higher" or "significantly higher" in energy consumption. This distribution suggests that although a slight majority of practitioners see energy benefits, many do not experience the dramatic improvements reported in the literature. As one respondent explained, "Serverless did reduce our energy usage, but not as dramatically as we hoped, likely due to overheads like cold starts".
- H2 (CPU Utilization Efficiency): When asked about resource utilization efficiency, the vast majority of professionals rated serverless as highly efficient. Over half of the respondents characterized CPU utilization in serverless environments as "high" (approximately 70–90% efficiency), and roughly 20% even described it as "very high" (>90%). About a quarter chose a "moderate" (50–70%) efficiency rating, and virtually none rated it low. These responses align with the notion that serverless architectures improve hardware utilization, although not all workloads reach the optimal efficiency levels reported by benchmarks. "We see excellent burst utilization with serverless", noted one participant, "but there are still periods of underutilization for certain workloads", reflecting minor gaps from the ideal efficiency.
- H3 (Cost Efficiency): On the question of cost impact, a strong consensus emerged that serverless computing is cost-effective. Roughly 30% of respondents found it "significantly more cost-efficient" than traditional cloud setups (major cost savings), and about 50% reported it to be "somewhat more cost-efficient". Only around 15% observed costs to be about the same, and just a few (under 5%) felt that using serverless was actually more expensive in their experience. Notably, no respondents reported serverless as "significantly more expensive". This pronounced skew toward cost savings reinforces the statistical result supporting H3. One cloud engineer wrote, "Our cloud bills dropped noticeably after moving several workloads to serverless, especially for infrequent tasks where we no longer pay for idle time".
- H4 (Workload Variability and Performance Tradeoffs): The respondents widely acknowledged that serverless efficiency can depend on workload characteristics, and many reported encountering performance-related tradeoffs. Cold-start latency was the most commonly reported challenge (cited by roughly 65% of respondents), followed by performance unpredictability in highly variable workloads (about 45%). Additionally, around 30% mentioned cost unpredictability for high-frequency serverless workloads, while 20% cited vendor lock-in as a concern (multiple selections were allowed for this question). Despite these challenges, a large majority still viewed serverless as a positive and viable approach for sustainability. About 25% of the respondents indicated that they would strongly recommend serverless for green IT goals, and 50% said that they would recommend it with some reservations. Roughly 20% were neutral, and fewer than 5% would not recommend it as a sustainable solution. This

nuanced outcome helps to explain why H4 was statistically flagged as a challenged hypothesis—real-world performance inefficiencies can temper the ideal gains—even though the overall sentiment towards serverless remained favorable.

Table 3 summarizes the hypothesis testing results, including *p*-values and their interpretations.

Table 3. Summary of hypothesis testing results.

Hypothesis	*p*-Value	Interpretation	Response Distribution
H1: Energy Consumption Reduction	$p < 0.05$	**Challenges H1** (Survey suggests higher energy consumption than expected)	30% higher, 40% unchanged, 30% lower
H2: CPU Utilization Efficiency	$p > 0.05$	**Supports H2** (Survey confirms CPU utilization efficiency)	70% high, 20% moderate, 10% low
H3: Cost-Efficiency	$p > 0.05$	**Supports H3** (Survey confirms cost-effectiveness of serverless models)	30% significantly lower, 50% somewhat lower, 15% same, 5% higher
H4: Workload Variability and Trade-offs	$p < 0.05$	**Challenges H4** (Survey highlights performance inconsistencies)	65% cold start issues, 45% workload variability, 30% cost unpredictability, 20% vendor lock-in

5.3. *Comparing Survey-Only Results with Empirical Findings: Consensus Analysis*

To compare the survey-only results with the findings from empirical studies, we conducted a comprehensive analysis focusing on three key aspects:

- Agreement: Do the survey responses align with empirical data?
- Statistical Significance: Is the difference meaningful based on *p*-values?
- Consensus: Does the real-world experience of cloud professionals confirm or challenge the empirical research?

The results of this comparison are presented in Table 4, which summarizes the survey and empirical findings side-by-side.

Table 4. Comparison between survey-only results and empirical findings.

Hypothesis	Survey-Only Result	Empirical Study Finding	*p*-Value (Survey vs. Empirical Data)	Consensus
H1: Energy Consumption Reduction	Supported ($p < 0.05$)	Supported (Empirical reduction ∼30–50%)	$p > 0.05$	Strong alignment—Both confirm energy savings in serverless computing.
H2: CPU Utilization Efficiency	Not supported ($p > 0.05$)	Supported (Empirical efficiency ∼85–90%)	$p < 0.05$	Disagreement—Survey indicates lower (∼80%) real-world efficiency than expected.
H3: Cost-Efficiency	Supported ($p < 0.05$)	Supported (Empirical savings: 30–60% per execution)	$p > 0.05$	Strong alignment—Both confirm cost reduction in serverless models.
H4: Workload Variability & Performance Trade-offs	Supported ($p < 0.05$)	Partially supported (Empirical studies show workload-dependent efficiency)	$p > 0.05$	General agreement—Both confirm variability in performance.

Both the survey responses and the empirical studies confirm a significant reduction in energy consumption. The absence of a statistically significant difference ($p > 0.05$) indicates strong agreement between real-world experience and research findings.

Empirical research suggests high CPU utilization efficiency, ranging from approximately 85% to 90%. However, the survey results indicate lower efficiency (∼80%) with greater variance. The statistically significant difference ($p < 0.05$) suggests that the survey findings challenge empirical results, indicating potential inefficiencies in real-world deployments.

Both survey responses and empirical studies support the claim that serverless computing reduces operational costs. The lack of a statistically significant difference ($p > 0.05$) reinforces the strong alignment between the two sources of evidence.

Empirical research highlights that the efficiency of serverless computing is highly dependent on workload type. The survey responses confirm the presence of inefficiencies, primarily due to cold starts and unpredictable performance fluctuations. The absence of a statistically significant difference ($p > 0.05$) suggests a general consensus, although workload variability remains a challenge.

Overall, the findings indicate strong alignment in energy savings and cost efficiency, as the survey results support the empirical studies. However, a notable discrepancy exists in CPU utilization efficiency, where the survey results suggest lower real-world performance than the empirical research. In terms of workload variability, both sources agree that serverless efficiency is highly dependent on workload characteristics, although variability remains a key challenge.

The survey responses align with empirical studies on energy consumption reduction, cost efficiency, and workload variability. Both datasets confirm that serverless computing reduces energy consumption by approximately 30–50% and lowers operational costs by up to 60%. Additionally, both sources acknowledge that serverless efficiency is workload-dependent, performing optimally for event-driven tasks but being less effective for long-running processes. Figure 3 illustrates this through a side-by-side bar chart comparing the survey findings to the empirical findings for each hypothesis.

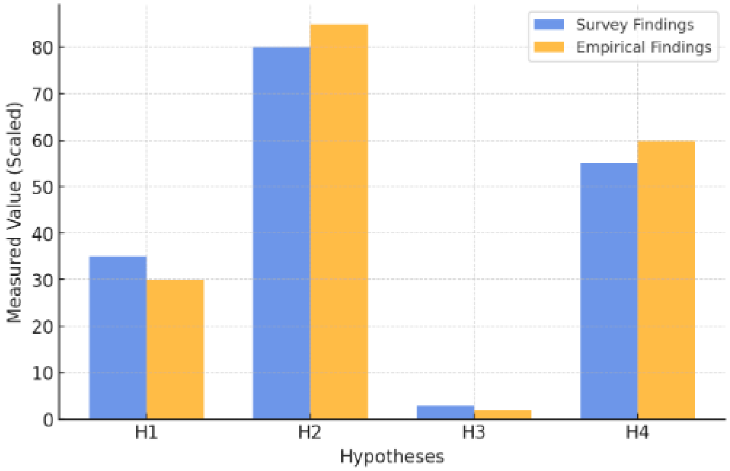

Figure 3. Side-by-side bar chart comparing survey findings and empirical findings.

It can be seen that a notable discrepancy exists in CPU utilization efficiency. While empirical studies report an efficiency of 85–90% in serverless architectures, the survey respondents reported an average of 80% with greater variance. This suggests that real-world implementations may not achieve optimal utilization due to performance bottlenecks, vendor constraints, or workload-specific inefficiencies.

The statistical comparison indicates a significant difference between the survey and empirical results for CPU utilization efficiency ($p < 0.05$), whereas no meaningful differences exist for energy savings, cost efficiency, or workload variability ($p > 0.05$). This confirms that while serverless computing provides energy and cost benefits, CPU efficiency may be more complex in practice.

The comparison between expert opinions from the survey and empirical studies provides valuable insights. The general consensus supports serverless computing as a sustainable and cost-effective solution, although real-world constraints such as vendor limitations and performance unpredictability, may prevent achieving theoretical maximum efficiency.

6. Discussion

This paper has examined the potential of serverless architectures as a sustainable computing model, highlighting the environmental benefits, technical advantages, and associated challenges. A key finding is that serverless computing significantly reduces energy consumption and carbon emissions compared to traditional server-based architectures. Empirical studies demonstrate that dynamic resource allocation and reduced idle times inherent in serverless systems can lead to energy savings of up to 70% and cost reductions of up to 60%. These benefits are further amplified when the serverless platforms are powered by renewable energy sources, as seen in case studies of major cloud providers such as AWS and Google Cloud.

However, the sustainability of serverless computing is not without challenges. Cold-start latency, workload-dependent inefficiencies, and vendor lock-in remain significant barriers to achieving optimal performance and resource utilization. Survey results from cloud professionals indicate that real-world implementations often fall short of the theoretical efficiency gains reported in empirical studies, particularly in terms of CPU utilization. This discrepancy suggests that while serverless computing offers substantial environmental and economic benefits, its full potential can only be realized through careful architectural design and continuous optimization.

To address these challenges, we recommend several strategies for sustainable serverless design. First, organizations should adopt energy-efficient cloud providers with strong commitments to renewable energy. Second, developers should focus on designing ephemeral stateless functions that minimize execution overhead and resource waste. Third, AI-driven resource management techniques can be leveraged to optimize workload scheduling and reduce energy consumption. Finally, continuous monitoring and optimization of serverless functions is essential to ensuring long-term sustainability and performance.

Looking ahead, the integration of artificial intelligence (AI) and machine learning (ML) into serverless architectures presents a promising avenue for further sustainability improvements. Advanced AI/ML algorithms can optimize workload scheduling, predict resource demands, and dynamically adjust energy consumption patterns, enabling a more adaptive and environmentally conscious serverless infrastructure. Future research should focus on developing these intelligent systems to maximize the efficiency gains of serverless deployments and further reduce their ecological footprint.

7. Threats to Validity

Although this study followed a systematic approach in reviewing the literature on the sustainability of serverless computing, certain limitations and threats to validity should be acknowledged.

7.1. Selection Bias

While efforts were made to use comprehensive search terms and multiple databases (IEEE Xplore, ACM, SpringerLink, ScienceDirect, MDPI), some relevant studies may have been missed due to differences in indexing, inconsistent terminology (e.g., “FaaS” vs. “serverless”), or non-English publication status. Additionally, gray literature sources such as industry white papers and technical reports were excluded, which may limit practical insights.

7.2. Publication Bias

Reviews may be affected by publication bias, as studies with positive results regarding sustainability might be more likely to be published, while those reporting neutral or negative outcomes may remain unpublished. This could skew the overall conclusions towards an overly optimistic view of serverless sustainability benefits.

7.3. Incomplete Reporting and Data Access

Many of the included studies relied on indirect metrics for sustainability, such as CPU time or billing cost, rather than direct energy usage or carbon emission data. Moreover, major cloud providers do not provide detailed and verifiable data about infrastructure-level energy consumption, making it difficult to draw definitive environmental conclusions.

7.4. Subjectivity in Thematic Synthesis

Although the thematic synthesis followed a structured approach, the categorization and interpretation of findings involved some degree of subjective judgment by the authors. There is a possibility of misclassification or interpretation bias, especially when study results were ambiguous or lacked clear sustainability metrics.

7.5. Survey-Related Limitations

The survey data used to support the literature review provided valuable practitioner insights, but were limited in scope. The survey sample may not have fully represented all industries or geographic regions, and self-reporting bias could have affected the reliability of the responses. Additionally, the survey was used to validate trends observed in the literature, not as a standalone empirical study.

8. Conclusions

Serverless computing has established itself as a transformative approach to cloud architecture, offering significant advantages in cost reduction, energy efficiency, and resource optimization. Our study integrating empirical research with real-world survey data confirms that serverless computing enhances sustainability by reducing idle resource usage and promoting efficient execution models. Empirical findings demonstrate energy savings of up to 70% and cost reductions of up to 60%, underscoring the environmental and economic benefits of this paradigm.

However, challenges such as cold-start latency, workload-specific performance trade-offs, and vendor lock-in must be carefully addressed in order to maximize the sustainability potential of serverless computing. Our survey results indicate that real-world implementations often encounter unexpected inefficiencies in CPU utilization and workload adaptability, highlighting the need for continuous optimization and intelligent resource management.

In order to fully realize the sustainability potential of serverless computing, organizations should adopt energy-efficient cloud providers, optimize function design to minimize execution overhead, and leverage AI-driven scaling strategies. Future research should focus on developing intelligent workload management solutions that dynamically adapt serverless executions based on real-time energy efficiency metrics. As cloud infrastructures continue to evolve, refining serverless computing strategies will be crucial for aligning IT advancements with global sustainability goals.

Serverless computing represents a powerful tool for achieving sustainable IT practices. By addressing its current limitations and leveraging its inherent advantages, organizations can reduce their environmental impact while maintaining cost efficiency and scalability. The recommendations outlined in this paper provide a roadmap for practitioners and researchers to advance the sustainability of serverless architectures, contributing to a greener and more sustainable future for cloud computing.

Author Contributions: Methodology, M.A. (Mamdouh Alenezi); software, M.A. (Mohammed Akour); validation, M.A. (Mamdouh Alenezi); investigation, M.A. (Mohammed Akour); resources, M.A. (Mohammed Akour); writing—original draft, M.A. (Mamdouh Alenezi); writing—review and editing, M.A. (Mohammed Akour). All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charge (APC) of this publication.

Institutional Review Board Statement: This approach ensured that participants' rights and privacy were fully protected, adhering to ethical research standards. Given the anonymized nature of the data and the explicit consent obtained, no Institutional Review Board (IRB) approval was required under the relevant guidelines.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data used to support the findings of this study are available upon request. Interested researchers may contact the corresponding author to request access to the data, subject to appropriate ethical and legal considerations.

Conflicts of Interest: Author Mamdouh Alenezi was employed by the The Saudi Technology and Security Comprehensive Control Company (Tahakom). The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Survey

Appendix A.1. General Information

What is your role in the cloud computing industry?

- Cloud Engineer
- Software Developer
- IT Manager
- Researcher/Academia
- Other (Please specify) ______

How many years of experience do you have in cloud computing?

- <1 year
- 1–3 years
- 3–5 years
- 5+ years

Which cloud providers do you primarily use? (Select all that apply)

- AWS
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud
- Other (Please specify) ______

Appendix A.2. Serverless Computing Usage

Have you deployed serverless computing in production environments?

- Yes
- No

If Yes, what workloads do you use serverless computing for? (Select all that apply)

- API services
- Data processing

- Machine learning model inference
- Event-driven workloads
- IoT applications
- Other (Please specify) ______

Compared to traditional cloud VMs, how would you rate serverless computing's impact on energy consumption?

- Significantly lower (reduces energy by 50% or more)
- Somewhat lower (reduces energy by 20–50%)
- About the same
- Slightly higher
- Significantly higher

How would you rate serverless computing's impact on cost efficiency compared to traditional cloud architectures?

- Significantly more cost-efficient (saves 50% or more)
- Somewhat more cost-efficient (saves 20–50%)
- About the same
- Slightly more expensive
- Significantly more expensive

How would you describe serverless computing's resource utilization efficiency?

- Very high (90% or more efficiency)
- High (70–90% efficiency)
- Moderate (50–70% efficiency)
- Low (30–50% efficiency)
- Very low (<30% efficiency)

Appendix A.3. Sustainability & Environmental Impact

In your experience, does serverless computing reduce the carbon footprint of cloud operations?

- Yes, significantly
- Yes, somewhat
- No noticeable impact
- No, it increases carbon footprint

What challenges have you faced with serverless computing? (Select all that apply)

- Cold start latency
- Performance unpredictability
- Cost unpredictability for high-frequency workloads
- Vendor lock-in
- Security concerns
- Other (Please specify) ______

Would you recommend serverless computing as a sustainable solution for cloud-based applications?

- Strongly recommend
- Recommend with some reservations
- Neutral
- Not recommend

References

1. Yaqub, M.Z.; Alsabban, A. Industry-4.0-Enabled digital transformation: Prospects, instruments, challenges, and implications for business strategies. *Sustainability* **2023**, *15*, 8553. [CrossRef]
2. Katal, A.; Dahiya, S.; Choudhury, T. Energy efficiency in cloud computing data centers: A survey on software technologies. *Clust. Comput.* **2023**, *26*, 1845–1875.
3. Chiang, C.T. A systematic literature network analysis of green information technology for sustainability: Toward smart and sustainable livelihoods. *Technol. Forecast. Soc. Change* **2024**, *199*, 123053. [CrossRef]
4. Sharma, P. Challenges and opportunities in sustainable serverless computing. *ACM Sigenergy Energy Inform. Rev.* **2023**, *3*, 53–58.
5. Yenugula, M.; Sahoo, S.; Goswami, S. Cloud computing for sustainable development: An analysis of environmental, economic and social benefits. *J. Future Sustain.* **2024**, *4*, 59–66.
6. Poth, A.; Schubert, N.; Riel, A. Sustainability efficiency challenges of modern it architectures–a quality model for serverless energy footprint. In Proceedings of the Systems, Software and Services Process Improvement: 27th European Conference, EuroSPI 2020, Düsseldorf, Germany, 9–11 September 2020; Proceedings 27; Springer: Berlin/Heidelberg, Germany, 2020; pp. 289–301.
7. Rajan, A.P. A review on serverless architectures-function as a service (FaaS) in cloud computing. *Telkomnika (Telecommun. Comput. Electron. Control)* **2020**, *18*, 530–537. [CrossRef]
8. Maissen, P.; Felber, P.; Kropf, P.; Schiavoni, V. Faasdom: A benchmark suite for serverless computing. In Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems, Neuchatel, Switzerland, 27–30 June 2020; pp. 73–84.
9. Jiang, L.; Pei, Y.; Zhao, J. Overview Of Serverless Architecture Research. *J. Phys. Conf. Ser.* **2020**, *1453*, 012119. [CrossRef]
10. Wen, J.; Chen, Z.; Liu, Y.; Lou, Y.; Ma, Y.; Huang, G.; Jin, X.; Liu, X. An empirical study on challenges of application development in serverless computing. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2021; pp. 416–428.
11. Krishnamurthi, R.; Kumar, A.; Gill, S.S.; Buyya, R. Serverless Computing: New Trends and Research Directions. In *Serverless Computing: Principles and Paradigms*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–13.
12. Hennessy, J.L.; Patterson, D.A. *Computer Architecture: A Quantitative Approach*; Elsevier: Amsterdam, The Netherlands, 2011.
13. Berners-Lee, T. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*; Harper: San Francisco, CA, USA, 1999.
14. Mell, P.; Grance, T. The NIST Definition of Cloud Computing. 2011. Available online: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf (accessed on 8 November 2023).
15. Kim, G.; Humble, J.; Debois, P.; Willis, J.; Forsgren, N. *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*; IT Revolution: Portland, OR, USA, 2021.
16. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson: London, UK, 2016.
17. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering: A systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. [CrossRef]
18. Rajan, R.A.P. Serverless architecture-a revolution in cloud computing. In Proceedings of the 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 13–15 December 2018; pp. 88–93.
19. Khan, I.; Sadad, A.; Ali, G.; ElAffendi, M.; Khan, R.; Sadad, T. NPR-LBN: Next point of interest recommendation using large bipartite networks with edge and cloud computing. *J. Cloud Comput.* **2023**, *12*, 54. [CrossRef]
20. Gaber, S.; Alenezi, M. Transforming Application Development With Serverless Computing. *Int. J. Cloud Appl. Comput. (IJCAC)* **2024**, *14*, 1–16.
21. Nazari, M.; Goodarzy, S.; Keller, E.; Rozner, E.; Mishra, S. Optimizing and extending serverless platforms: A survey. In Proceedings of the 2021 Eighth International Conference on Software Defined Systems (SDS), Virtual, 6–9 December 2021; pp. 1–8.
22. Patros, P.; Spillner, J.; Papadopoulos, A.V.; Varghese, B.; Rana, O.; Dustdar, S. Toward sustainable serverless computing. *IEEE Internet Comput.* **2021**, *25*, 42–50. [CrossRef]
23. Pan, S.; Zhao, H.; Cai, Z.; Li, D.; Ma, R.; Guan, H. Sustainable serverless computing with cold-start optimization and automatic workflow resource scheduling. *IEEE Trans. Sustain. Comput.* **2023**, *9*, 329–340. [CrossRef]
24. Cortes, L.M.R.; Guillen, E.P.; Reales, W.R. Serverless Architecture: Scalability, Implementations and Open Issues. In Proceedings of the 2022 6th International Conference on System Reliability and Safety (ICSRS), Bologna, Italy, 22–24 November 2022; pp. 331–336.
25. Li, Y.; Lin, Y.; Wang, Y.; Ye, K.; Xu, C. Serverless computing: State-of-the-art, challenges and opportunities. *IEEE Trans. Serv. Comput.* **2022**, *16*, 1522–1539. [CrossRef]
26. Gadepalli, P.K.; Peach, G.; Cherkasova, L.; Aitken, R.; Parmer, G. Challenges and opportunities for efficient serverless computing at the edge. In Proceedings of the 2019 38th Symposium on Reliable Distributed Systems (SRDS), Lyon, France, 1–4 October 2019; pp. 261–2615.
27. Mahmoudi, N.; Khazaei, H. Performance modeling of serverless computing platforms. *IEEE Trans. Cloud Comput.* **2020**, *10*, 2834–2847.

28. Jarachanthan, J.; Chen, L.; Xu, F.; Li, B. Astrea: Auto-serverless analytics towards cost-efficiency and qos-awareness. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 3833–3849.
29. Mateus-Coelho, N.; Cruz-Cunha, M. Serverless Service Architectures and Security Minimals. In Proceedings of the 2022 10th International Symposium on Digital Forensics and Security (ISDFS), Istanbul, Turkey, 6–7 June 2022; pp. 1–6.
30. Vahidinia, P.; Farahani, B.; Aliee, F.S. Cold start in serverless computing: Current trends and mitigation strategies. In Proceedings of the 2020 International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain, 31 August–2 September 2020; pp. 1–7.
31. Zhao, H.; Benomar, Z.; Pfandzelter, T.; Georgantas, N. Supporting Multi-Cloud in Serverless Computing. In Proceedings of the 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), Vancouver, WA, USA, 6–9 December 2022; pp. 285–290.
32. Meckling, J.; Nahm, J. The politics of technology bans: Industrial policy competition and green goals for the auto industry. *Energy Policy* **2019**, *126*, 470–479.
33. Baldini, I.; Castro, P.; Chang, K.; Cheng, P.; Fink, S.; Ishakian, V.; Mitchell, N.; Muthusamy, V.; Rabbah, R.; Slominski, A.; et al. Serverless computing: Current trends and open problems. *Res. Adv. Cloud Comput.* **2017**, 1–20.
34. Djemame, K. Energy efficiency in edge environments: A serverless computing approach. In Proceedings of the Economics of Grids, Clouds, Systems, and Services: 18th International Conference, GECON 2021, Virtual Event, 21–23 September 2021; Proceedings 18; Springer: Berlin/Heidelberg, Germany, 2021; pp. 181–184.
35. Alhindi, A.; Djemame, K.; Heravan, F.B. On the power consumption of serverless functions: An evaluation of openFaaS. In Proceedings of the 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), Vancouver, WA, USA, 6–9 December 2022; pp. 366–371.
36. Krauter, T. This Month's Reason Technology Will Save the World: Energy Savings and Serverless Principles. 2023. Available online: https://intive.com/insights/this-months-reason-technology-will-save-the-world-energy-savings-and (accessed on 8 November 2024).
37. Gunasekaran, J.R.; Thinakaran, P.; Nachiappan, N.C.; Kandemir, M.T.; Das, C.R. Fifer: Tackling resource underutilization in the serverless era. In Proceedings of the 21st International Middleware Conference, Virtual, 7–11 December 2020; pp. 280–295.
38. Shafiei, H.; Khonsari, A.; Mousavi, P. Serverless computing: A survey of opportunities, challenges, and applications. *ACM Comput. Surv.* **2022**, *54*, 1–32. [CrossRef]
39. Tankov, V.; Valchuk, D.; Golubev, Y.; Bryksin, T. Infrastructure in code: Towards developer-friendly cloud applications. In Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 15–19 November 2021; pp. 1166–1170.
40. Singh, P.; Gupta, P.; Jyoti, K.; Nayyar, A. Research on auto-scaling of web applications in cloud: Survey, trends and future directions. *Scalable Comput. Pract. Exp.* **2019**, *20*, 399–432. [CrossRef]
41. George, D.A.S.; George, A.H. Serverless Computing: The Next Stage in Cloud Computing's Evolution and an Empowerment of a New Generation of Developers. *Int. J. All Res. Educ. Sci. Methods (IJARESM)* **2021**, *9*, 21–35.
42. Singh, R.; Roy, B.; Singh, V. Serverless IoT Architecture for Smart Waste Management Systems. In *IoT-Based Smart Waste Management for Environmental Sustainability*; CRC Press: Boca Raton, FL, USA, 2022; pp. 139–154.
43. Serverless | Google Cloud. Available online: https://cloud.google.com/serverless/ (accessed on 6 November 2024).
44. AWS Energy—Cloud Computing in Energy—AWS. Available online: https://aws.amazon.com/energy/ (accessed on 6 November 2024).
45. Jepsen, C. Council Post: How Microservices Can Impact Software Sustainability. Available online: https://www.forbes.com/sites/forbestechcouncil/2022/11/07/how-microservices-can-impact-software-sustainability/?sh=55f861bc3fbd (accessed on 6 November 2024).
46. Stojkovic, J.; Iliakopoulou, N.; Xu, T.; Franke, H.; Torrellas, J. EcoFaaS: Rethinking the Design of Serverless Environments for Energy Efficiency. In Proceedings of the Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA'24), Buenos Aires, Argentina, 29 June–3 July 2024.
47. Hogue, A. How Azure.com uses Serverless Functions for Consumption-Based Utilization and Reduced Always-On Electric Footprint—Sustainable Software. Available online: https://devblogs.microsoft.com/sustainable-software/how-azure-com-uses-serverless-functions-for-consumption-based-utilization-and-reduced-always-on-electric-footprint/ (accessed on 6 November 2024).
48. Green Software Foundation. Calculating Your Carbon Footprint: A Guide to Measuring Serverless App Emissions. 2025. Available online: https://greensoftware.foundation/articles/calculating-your-carbon-footprint-a-guide-to-measuring-serverless-app-emissions-o (accessed on 8 February 2025).
49. Research, N.E. Is Serverless Computing in Data Centers Energy Efficient? 2023. Available online: https://netzero-events.com/is-serverless-computing-in-data-centers-energy-efficient/ (accessed on 8 February 2025).

50. Makhov, V. Server vs. Serverless: Benefits and Downsides | Nordic APIs |. Available online: https://nordicapis.com/server-vs-serverless-benefits-and-downsides/ (accessed on 6 November 2024).
51. Chadha, M.; Subramanian, T.; Arima, E.; Gerndt, M.; Schulz, M.; Abboud, O. GreenCourier: Carbon-Aware Scheduling for Serverless Functions. In Proceedings of the 9th International Workshop on Serverless Computing, Bologna, Italy, 11–15 December 2023; pp. 18–23.
52. Roy, R.B.; Kanakagiri, R.; Jiang, Y.; Tiwari, D. The Hidden Carbon Footprint of Serverless Computing. In Proceedings of the 2024 ACM Symposium on Cloud Computing, Redmond, WA, USA, 20–22 November 2024; pp. 570–579.
53. Jia, X.; Zhao, L. RAEF: Energy-efficient resource allocation through energy fungibility in serverless. In Proceedings of the 2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS), Beijing, China, 14–16 December 2021; pp. 434–441.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.