Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

Systematic Review and Analysis of Cost-Saving Mechanisms, Challenges, And Best Practices in A Serverless Computing Environment

***Benjamin Asubam Weyori[1], Abdul Karim Mohammed[2], Samuel Gbli Tetteh[3]**

*[1]Department of Computer and Electrical Engineering, University of Energy and Natural Resources, Sunyani, Ghana,** benjamin.weyori@uenr.edu.gh

[2]Department of Computer Science and Informatics, University of Energy and Natural Resources Sunyani, Ghana,

[3]D Jarvis College of Computing and Digital Media, DePaul University, Chicago, USA

Abstract

Serverless computing is a cloud computing execution model enabling developers to concentrate more on business logic than infrastructure or server management. Developers and companies alike are drawn to this new paradigm because it minimises and completely removes the overhead associated with infrastructure, scaling, and provisioning. Given how new this phenomenon is, it is encouraging to think optimistically about serverless computing's potential acceptance. The study aims to offer an extensive understanding of the cost-saving mechanism in a serverless computing environment, as well as challenges and best practices to aid decision-making and optimisation. Leveraging a systematic approach, relevant literature was scrutinised to extract valuable insights. Findings reveal a need for more literature on cost-saving mechanisms specific to serverless computing, highlighting the need for further exploration in this domain. Challenges identified encompass technical complexities, operational hurdles, and security concerns, underscoring the multifaceted nature of serverless environments. The analysis explores cost-saving mechanisms and showcases efficient resource utilisation, workload optimisation, and performance enhancement strategies. Best practices elucidate the significance of architectural design, scalability considerations, and performance monitoring in achieving cost efficiencies. The synthesis of findings culminates in actionable recommendations for practitioners and researchers, emphasising the importance of informed decision-making in adopting serverless solutions. The study underscores the evolving nature of serverless computing and the imperative of addressing cost dynamics to harness its full potential. By shedding light on uncharted territories and offering strategic insights, this review advances knowledge in serverless computing.

Keywords: Serverless Computing, Serverless Environment, Cloud Computing, Business Logic, Server Management, Workload Optimization

1. Introduction

Serverless computing is a technology that gives total control over the container services executed on demand to fulfill request to the cloud provider. In this way, these designs function as event-driven computations and eliminate the requirement for constantly operating systems. (Stigler 2018)

Serverless computing has become a revolutionary paradigm that provides a cost-effective and scalable method for application development and deployment (Castro et al., 2019). This model builds on the promises of cloud computing by enabling organizations to pay only for the resources they use, with virtually infinite scalability, while abstracting the complexities of server management (Jonas et al., 2019). As a result, The use of serverless computing has grown significantly in both research and industry, revolutionizing the way applications are built and operated.

Effective cost management plays a pivotal role in optimizing resource utilization and ensuring economic efficiency in serverless environments (Mishra et al., 2020). Given the dynamic nature of serverless platforms, understanding cost-saving mechanisms, addressing challenges, and implementing best practices are essential for organizations to leverage the full potential of serverless computing while controlling operational expenses (Castro et al., 2019).

This systematic review and analysis delve into cost-saving mechanisms, challenges, and best practices within serverless computing environments. By synthesising current literature and empirical studies, this research aims to offer a thorough knowledge of the landscape, offering insights that can guide decision-making and resource optimisation in serverless deployments. The objectives of this study encompass identifying key cost-saving strategies, elucidating prevalent challenges, and outlining recommended best practices to enhance cost efficiency and operational effectiveness in serverless environments.

Through a structured approach, this systematic review intends to add to the existing knowledge base on serverless computing, shedding light on critical aspects related to cost dynamics and resource management. By delineating the scope of the review and setting clear objectives, this study endeavours to offer valuable insights that can inform practitioners, researchers, and decision-makers in navigating the complexities of cost optimisation in serverless computing environments.

2. Literature Review

Serverless computing, a cloud computing model, abstracts server management from developers, allowing them to focus solely on code execution without the need to provision or manage servers (Baldini et al., 2017). This model operates on a pay-as-you-go basis, where users are billed based on actual usage rather than pre-allocated resources, offering scalability and cost-efficiency (Baldini et al., 2017). Key characteristics of serverless computing include event-driven architecture, auto-scaling, and stateless functions, enabling rapid development and deployment of applications (Baldini et al., 2017).

Research by emphasizes the dual objectives of cost reduction and elasticity in serverless computing systems (Ebrahimpour et al., 2023). Various studies have explored cost-saving mechanisms such as efficient function scheduling to balance cold start time and cost (Ebrahimpour et al., 2022), and workflow-aware analytical models for predicting performance and cost in serverless executions (Kumari et al., 2023). Additionally, best practices have been identified, including architectural design considerations and scalability strategies to optimize cost efficiency (Mampage et al., 2022).

Despite advancements in understanding cost dynamics in serverless environments, gaps persist in the literature. For instance, while studies have focused on resource management and performance prediction (Mahmoudi & Khazaei, 2022), there needs to be more comprehensive research on specific cost-saving mechanisms tailored to serverless computing (Mampage et al., 2022). Furthermore, integrating green practices in serverless environments and their impact on cost efficiency still needs to be explored (Alsheyadi et al., 2019). The literature also lacks a holistic view on efficiency improvement aspects such as approximate computing in the context of serverless platforms (Denninnart, 2023).

One significant aspect highlighted by Baldini et al. (2017) is the evolution of cloud programming models towards serverless computing, which offers scalability, cost-effectiveness, and operational simplicity. This evolution represents a shift in cloud technologies, emphasizing serverless platforms' maturity and wide adoption.

Kumari et al. (2023) proposes a heuristic optimisation algorithm to identify the optimal resource configuration for achieving the best response time within a specified budget constraint. This approach underscores the importance of efficient resource allocation in cost-saving strategies in serverless environments.

Moreover, Müller et al. (2020) analyze the economic and performance perspectives of serverless computing, demonstrating scenarios where serverless deployments are economically viable and offer performance benefits.

Understanding the financial implications of serverless architectures is crucial for organisations seeking to optimize costs effectively.

This review synthesizes the existing literature and highlights the need for further research to address these gaps and enhance the understanding of cost-saving mechanisms, challenges, and best practices in serverless computing environments.

Methodology

This paper systematically reviews and analyses cost-saving mechanisms, challenges, and best practices in a serverless computing environment.

Data collection

Using keywords: cost-saving mechanisms, serverless computing, Challenges in serverless computing, nature of serverless environments, a total of hundred and thirty (130) research papers related to the studies that were published in conferences, magazines and journals were obtained from the internet, whereas ten (10) documents in total were from other sources.

Study structure

David Moher contends that a protocol outlining the purpose, hypothesis, and intended procedures of the review should form the foundation of any systematic review; sadly, few systematic reviews include research reports on their study frameworks. (Moher et. Al.,2009) This study used the "PRISMA" (Preferred Reporting Items for Systematic Review and Meta-Analysis) methodology. This well-structured, detailed systematic review makes it easier to comprehend and assess researchers' methodologies. It lists all the research found, along with the inclusion and exclusion criteria and their justifications. There are five phases in the PRISMA model;

Phase 1: creating the "inclusion and exclusion" criteria, the research scope, and the competency question.
Phase 2 involves locating relevant terms in the literature.
Phase 3: Assessing the articles' applicability in cases where the abstracts satisfy the inclusion and exclusion requirements.
Phase 4 involves mapping keywords to describe the documents.
The meta-analysis of the review's studies is phase five.

After conducting a full-text exclusion, which removed twenty (20) duplicates and an additional fifteen (15) articles after additional exclusion criteria were applied, our online literature search turned up one hundred thirty (130) and ten (10) from other sources. This left the primary studies' refined lists at ninety-five (95). The following digital libraries are used: Web of Science, IEEE, Science Direct, Springer Open, and Research Gate. Figure 1 describes the systematic procedures that were used.

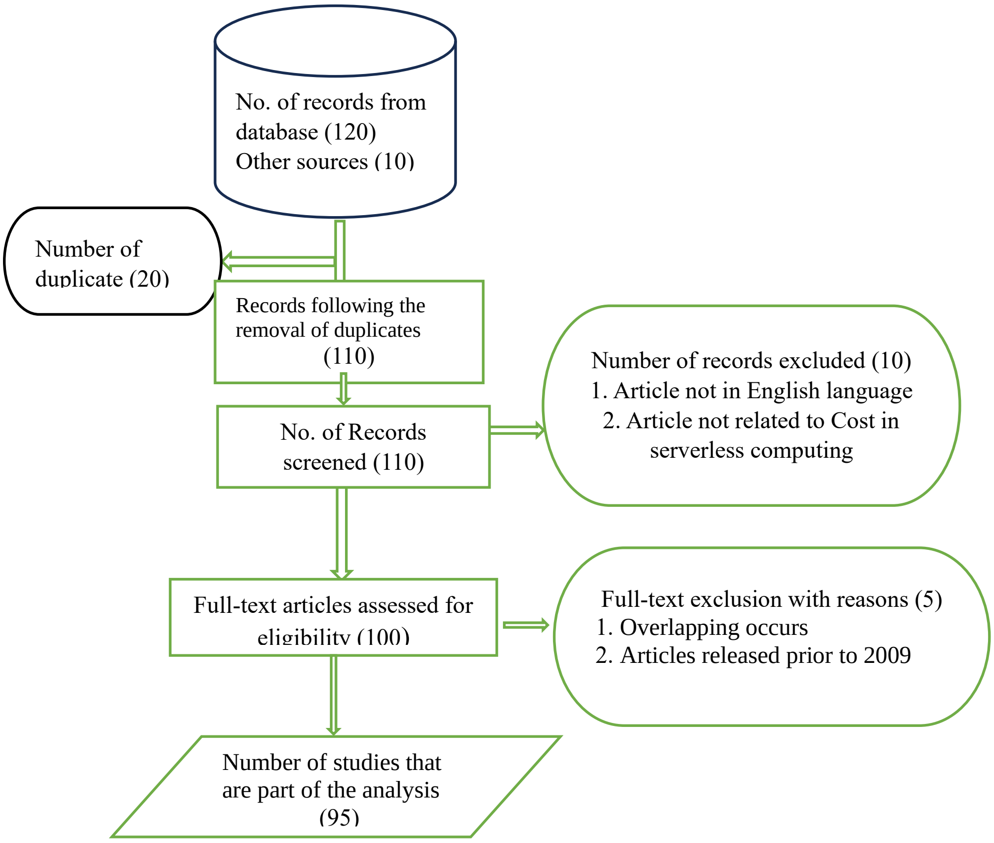

No. of records from database (120)
Other sources (10)
Number of duplicate (20)
Records following the removal of duplicates (110)
No. of Records screened (110)
Number of records excluded (10)
1. Article not in English language
2. Article not related to Cost in serverless computing
Full-text articles assessed for eligibility (100)
Full-text exclusion with reasons (5)
1. Overlapping occurs
2. Articles released prior to 2009
Number of studies that are part of the analysis (95)

Fig. 1 Article selection procedure (PRISMA)

Types of cost- saving mechanism

Pay-Per-Use Pricing Models

Pay-per-use pricing Models in serverless computing offer a cost-saving mechanism by charging users based on real resource usage instead of capacity reserved (Jiang et al., 2021). This model contrasts with traditional cloud computing, where people pay even on

reserved resources, potentially leading to cost inefficiencies (Kiener et al., 2021). By adopting a pay-per-use model, serverless platforms enable users to optimise costs by batching requests for processing, thereby increasing efficiency and reducing monetary expenses (Ali et al., 2022). This pricing mechanism aligns with a broader trend in the industry, where firms are transitioning from traditional product selling to selling product usage, as seen in the Pay-Per-Use (PPU) service model (Ladas et al., 2022).

The pay-per-use model in serverless computing is particularly advantageous for workloads that benefit from high parallelism, such as distributed training in machine learning (Liu et al., 2022). However, it is essential to consider workload characteristics, as high data traffic can increase costs and potentially negate the benefits of the pay-per-use model (Wang, 2020). Despite the appeal of pay-per-use pricing, Conflicting goals between service providers and customers may prevent it from being applied effectively. (Qu & He, 2014).

Overall, pay-per-use pricing in serverless computing aligns with the industry's shift towards more efficient and flexible cost structures. By charging based on actual resource consumption, this model promotes cost optimisation, scalability, and operational simplicity, making it a compelling choice for organisations looking to leverage cloud services efficiently.

Auto-Scaling:

Auto-scaling in serverless computing refers to the capability of serverless platforms to automatically modify an application's resource allocation by the demands of the task at hand. This feature allows serverless applications to manage different traffic volumes without human intervention, guaranteeing peak performance and cost-effectiveness (Yu et al., 2020). Serverless computing platforms achieve auto-scalability by dynamically provisioning resources in response to changes in demand, enabling users to scale their applications seamlessly without the need to manage servers or worry about resource allocation (Wu et al., 2021).

The auto-scaling property of serverless computing platforms is a key advantage that differentiates them from traditional cloud deployment models. In serverless environments, the platform provider manages the scaling process, allowing programmers to concentrate on developing code instead of worrying about infrastructure management (Shafiei et al., 2022). This automated scaling feature and a pay-as-you-go system offer developers a cost-effective solution for deploying applications that can efficiently handle fluctuating workloads (Eismann et al., 2021).

Furthermore, auto-scaling in serverless computing is closely tied to the event-driven nature of these platforms. Serverless runtimes are designed to scale resources based on the number of incoming requests, ensuring that applications can respond dynamically to changes in traffic patterns (Aytekin & Johansson, 2019). This event-driven architecture and auto-scaling capabilities make serverless computing an attractive option for applications with unpredictable workloads or varying processing requirements.

auto-scaling in serverless computing enables efficient resource management, cost optimisation, and seamless scalability for modern applications. By automatically adjusting resource allocation based on workload demands, serverless platforms empower developers to build flexible and responsive applications without the burden of manual scaling and infrastructure management.

Granular Billing

Granular billing is a detailed approach to billing that involves breaking down charges into small, specific components rather than lumping them together under broader categories. This method allows for a more precise and accurate representation of the services and associated costs. Granular billing is critical in various fields, such as healthcare, where detailed billing codes can accurately reflect the services rendered (Bakken et al., 2000). In healthcare, granular billing codes derived from longitudinal healthcare notes offer a more detailed and unbiased representation than traditional billing codes, which are often less specific (Ho et al., 2016).

Furthermore, in service-oriented architectures, granular billing plays a crucial role. Services can vary from essential functions to composite services that perform more complex tasks, such as specialised product billing applications (Luthria & Rabhi, 2009). By breaking down billing into granular components, organisations can better track costs, allocate resources efficiently, and understand the true expenses associated with each service provided.

Granular billing is also essential in sectors like energy consumption monitoring, where having fine temporal granularity of consumption data allows for implementing energy-saving schemes, reducing consumer bills, and better planning of energy networks (Gougeon et al., 2022). This detailed approach to billing benefits service providers, organisations, and consumers by providing a clearer understanding of their usage patterns and associated costs.

Granular billing is crucial in various industries to ensure accurate cost representation, efficient resource allocation, and improved decision-making processes.

Serverless-Specific Optimization Techniques

Various techniques can be employed to address challenges to optimise serverless functions, such as minimising cold start times and leveraging serverless features like provisioned concurrency. One approach is through snapshotting, where a fully-booted function image is stored on disk to reduce cold-start latency (Ustiugov et al., 2021). Additionally, function fusion can reduce latency during cold starts by considering parallel run scenarios (Lee et al., 2021). Furthermore, dynamic task placement can optimise edge-cloud serverless platforms' performance (Das et al., 2020).

To address the cold start issue, it is crucial to understand the underlying architecture of serverless platforms and the impact of interference from load over time (Kelly et al., 2020). Moreover, using a cold start mode can automatically destroy containers after executing an application, which can help manage resources efficiently (Benedetti et al., 2021)—additionally, investigating how public cloud systems are affected by infrastructure provisioning and how performance varies depending on the condition of the underlying virtual machine or container (Mohanty et al., 2018).

Furthermore, to enhance serverless performance, it is essential to consider the implications of container and package initialisation on cold start times (Rajput et al., 2022). Additionally, exploring RDMA-codesigned remote forks for serverless computing can offer fast execution without provisioned concurrency (Wei et al., 2022). By understanding these optimisation techniques and leveraging insights from research on serverless computing, developers can enhance the efficiency and performance of their serverless applications.

Workload management strategies are crucial in optimising costs in serverless computing environments. By effectively parallelising serverless workloads, significant cost savings can be achieved. demonstrated cost savings of up to 81% for AWS Lambda, 49% for GCF, and 69.8% for GCR through workload parallelisation (Kiener et al., 2021).

Additionally, workload batching can enhance response times and cost-effectiveness, particularly for machine learning serving workloads (Mahmoudi & Khazaei, 2022). introduced FUNCPIPE, a pipelined serverless framework that achieved up to 77% cost savings and 2.2X speedup compared to existing serverless-based frameworks (Liu et al., 2022).

Serverless computing offers advantages such as unlimited elasticity, pay-per-use models, and reduced start-up overhead, making it beneficial for data management workloads (Jiang et al., 2021).

Furthermore, the ability of serverless computing to scale costs down to zero during low-traffic periods is a crucial advantage for cost-saving (Copik et al., 2022).

By leveraging serverless frameworks' elasticity and fine-grain pricing, workloads like stream processing can benefit significantly (Choi, 2019). To ensure accurate performance evaluation and cost optimisation, it is essential to consider workload characteristics and system design. highlighted the importance of adapting serverless platforms to different workload needs to improve performance and cost-effectiveness (Mahmoudi & Khazaei, 2020; Mahmoudi & Khazaei, 2022).

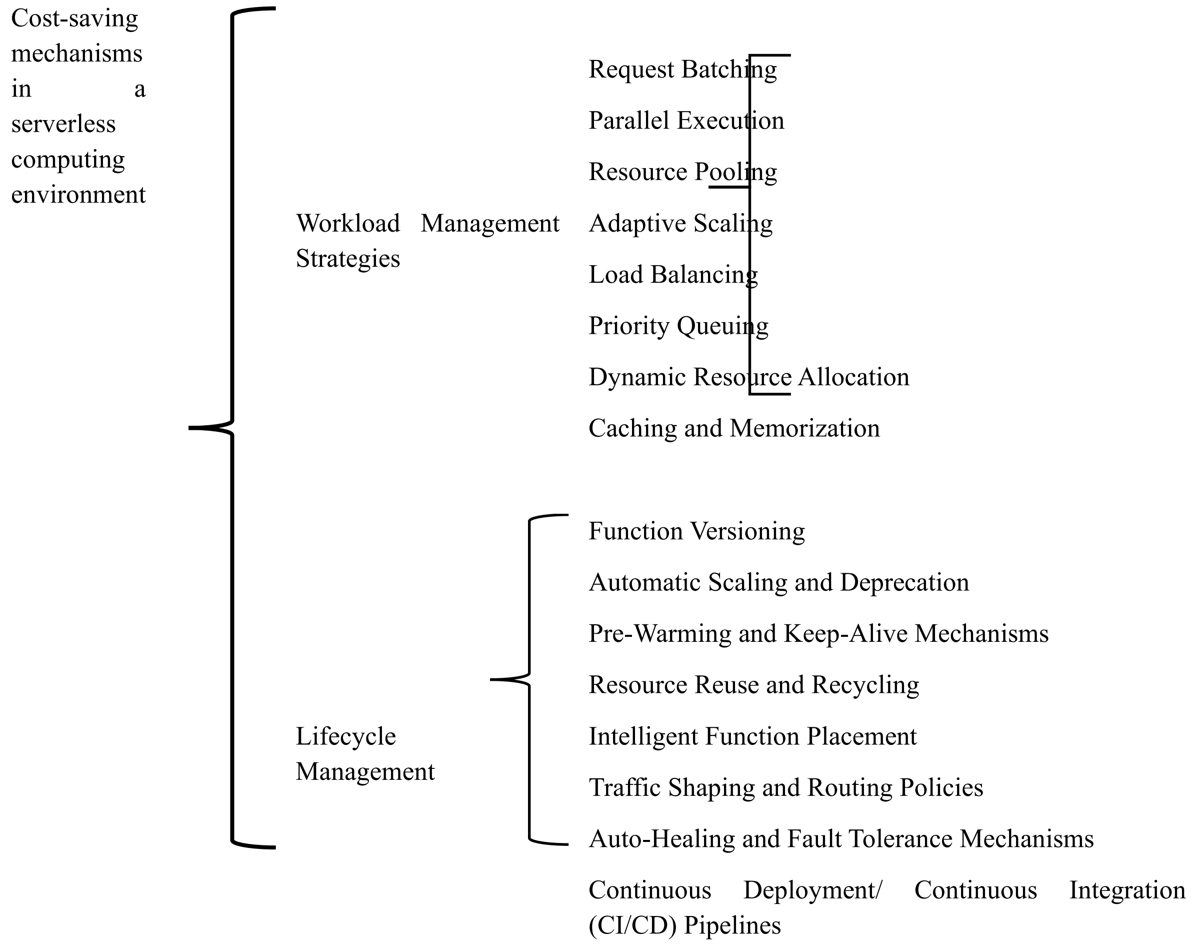
Additionally, accurate resource prediction and provisioning are crucial for maintaining cost-efficiency in dynamic serverless environments (Wang et al., 2021). In conclusion, implementing workload management strategies such as parallelisation, batching, and accurate resource prediction is essential for achieving cost savings in serverless computing environments. By optimising workload execution and resource allocation based on workload characteristics, organisations can effectively manage costs while ensuring high performance and scalability.

Pay-per-Use Pricing Model

Auto-Scaling

Granular Billing

Serverless-Specific Optimization Techniques



Cost-saving mechanisms in a serverless computing environment

Workload Management Strategies

Request Batching
Parallel Execution
Resource Pooling
Adaptive Scaling
Load Balancing
Priority Queuing
Dynamic Resource Allocation
Caching and Memorization

Lifecycle Management

Function Versioning
Automatic Scaling and Deprecation
Pre-Warming and Keep-Alive Mechanisms
Resource Reuse and Recycling
Intelligent Function Placement
Traffic Shaping and Routing Policies
Auto-Healing and Fault Tolerance Mechanisms
Continuous Deployment/ Continuous Integration (CI/CD) Pipelines

Fig. 2 Classifications of cost-saving mechanisms

Lifecycle management in a serverless computing environment can serve as a cost-saving mechanism by efficiently handling the entire lifecycle of functions. Organisations can optimise resource utilisation and minimise unnecessary costs by effectively managing the deployment, scaling, monitoring, and retirement of functions. proposed a container lifecycle-aware scheduling strategy for serverless computing, which can enhance resource allocation efficiency and reduce operational expenses (Wu et al., 2021) and additionally emphasised the importance of DevOps practices in managing serverless solutions across their lifecycle, ensuring efficient deployment and maintenance, which can contribute to cost savings (Palma et al., 2021).

Implementing appropriate lifecycle management techniques can lead to significant cost reductions. Organisations can avoid unnecessary expenses by automating resource adjustment in response to demand and efficiently handling function retirement. Furthermore, it highlighted the challenges posed by the fine granularity of function abstraction in serverless platforms, emphasising the need for comprehensive lifecycle management to optimise performance and cost-effectiveness (Eyk et al., 2020).

Moreover, the technique of instance pre-warming was discussed to lessen the cold start problem in serverless applications. Organisations can improve response times and potentially reduce costs by starting a predetermined number of function instances in advance and keeping them alive throughout the application lifecycle (Moreno-Vozmediano et al., 2022). This proactive approach to managing function instances can enhance performance while ensuring cost efficiency.

lifecycle management in serverless computing environments is crucial in optimising costs by efficiently handling functions' deployment, scaling, monitoring, and retirement. Organisations can achieve cost savings through improved resource utilisation and performance optimisation by implementing container lifecycle-aware scheduling, DevOps practices, and instance pre-warming.

4. Impact of overall cost on various models

Different pricing models in serverless computing, such as per-function invocation, memory size, and execution time, significantly impact overall costs. The fine-grained cost model in serverless platforms makes them cost-effective for tasks with short execution times and sporadic invocation patterns (Wawrzoniak et al., 2022). This is further supported by the fact that in serverless computing, customers pay for the processing resources they consume per second instead of the coarse-grained pricing associated with virtual machines and containers. (Patros et al., 2021).

Moreover, the choice of pricing model affects the general effectiveness and financial viability of serverless functions. For instance, when a function is started from a snapshot, its execution time is noticeably longer than when it is memory-resident, indicating a correlation between memory allocation and cost efficiency (Ustiugov et al., 2021). Additionally, a study by Mahmoudi & Khazaei highlights that when deploying machine learning to provide workloads on serverless systems, the absence of permanent state, small memory size, and restricted execution duration are the main limiting issues. (Mahmoudi & Khazaei, 2022).

The pricing models in serverless computing play a crucial role in determining the overall costs, performance, and efficiency of executing functions. Understanding how different pricing factors, such as invocation, memory, and execution time, impact costs is essential for optimising resource allocation and achieving cost-effective serverless deployments.

4.2 Serverless Computing Challenges

Cost management in serverless computing presents several key challenges that impact the overall cost-effectiveness of deployments. These challenges stem from the dynamic nature of serverless platforms and the complexities associated with optimising resource allocation and utilisation. By identifying and analysing these challenges, organisations can better navigate the cost implications of serverless environments and enhance operational efficiency.

challenges in cost-saving mechanisms in a serverless computing environment concerning various aspects, along with their corresponding solutions and suggested best practices:

Challenge	**Dynamic Workload Management**	**Granular Billing Complexity**	**Serverless-Specific Optimization**	**Workload Management Complexity**	**Lifecycle Management Efficiency**
Description	Managing dynamic workloads efficiently to optimize cost savings is challenging due to fluctuations in demand, leading to underutilization or over-	Granular billing models in serverless environments, where users are charged based on resource consumption at a fine-grained level, can lead to complexities in optimizing	Serverless computing introduces unique optimization challenges due to its event-driven and ephemeral nature, requiring specialized optimization techniques.	Balancing performance requirements with cost constraints in workload management is complex, requiring efficient workload distribution, prioritization, and scheduling.	Inefficient lifecycle management practices, including resource creation, execution, and termination, can result in resource wastage and increased costs.

	provisioning of resources	cost-saving strategies.			
Solution	Implementing intelligent auto-scaling mechanisms that dynamically adjust resources based on workload patterns can help optimize cost savings by ensuring optimal resource utilization.	Developing automated tools and algorithms to analyze granular billing data, this difficulty can be lessened by locating cost-saving options and allocating resources optimally based on cost-efficiency measures. .	Research and develop serverless-specific optimization techniques that take into account factors such as cold start latency, execution duration, and resource utilization patterns to optimize cost-saving mechanisms effectively.	Implementing intelligent workload management strategies that consider factors such as workload characteristics, resource availability, and cost objectives can help optimize cost-saving mechanisms in a serverless environment.	Implementing lifecycle management policies and automation techniques to efficiently manage serverless resources throughout their lifecycle can optimize cost-saving mechanisms.
Best Practices	Monitor workload patterns and adjust resource allocation dynamically in response to changes in demand. Use predictive analytics to forecast workload trends and proactively scale resources to meet anticipated demand. Implement auto-scaling policies based on performance metrics, such as latency or	Utilize cost analysis tools provided by cloud providers to monitor and analyze resource usage and costs. Implement resource tagging and labeling to track costs associated with specific services or applications accurately. Regularly review and optimize resource allocation based on cost-performance	Optimize function execution times to minimize cold start latency and reduce resource consumption. Use warm-up strategies to pre-load functions and reduce cold start times. Implement resource pooling and reuse strategies to minimize resource allocation overhead and improve efficiency.	Prioritize critical workloads based on business priorities and allocate resources accordingly. Implement workload-aware scheduling algorithms to optimize resource utilization and minimize costs. Utilize workload orchestration tools to automate workload deployment, scaling, and management based on predefined policies and SLAs.	Implement auto-scaling policies based on workload demand to dynamically adjust resources. Use resource pooling and recycling techniques to maximize resource reuse and minimize resource allocation overhead. Automate resource provisioning and deprovisioning processes to

	throughput, to maintain service quality while optimizing costs.	trade-offs to maximize cost savings.			reduce manual intervention and optimize resource utilization.

Table 1: Challenges and Best Practice in Serverless Computing.

4.3. **Real-World Serverless Deployments**

In a real-world case study by Christidis et al. (2020), the implementation of extensive AI workloads in an intelligent transportation system showcased the potential of serverless computing in enabling projections of train movements in real-time, on-demand for the UK rail network. The study demonstrated the practical application of serverless platforms in handling complex AI workloads efficiently and cost-effectively.

Analysing cost trends, resource utilisation patterns, and optimisation outcomes in serverless deployments is crucial for assessing the impact on operational efficiency and cost-effectiveness. Kumari et al. (2022) presented case research demonstrating serverless computing's application in healthcare systems, showcasing how serverless deployments can streamline operations, improve resource utilisation, and optimise costs in healthcare applications. By leveraging serverless platforms, healthcare systems can achieve scalability, flexibility, and cost efficiency in managing critical workloads.

Eyk et al. (2020) conducted empirical analyses focused on developing a comprehensive serverless benchmark to evaluate the performance cost associated with executing serverless workloads on cutting-edge systems. Through real-world experiments, the study demonstrated how benchmarking can help compare performance metrics and optimise cost-effectiveness in serverless computing environments.

Furthermore, Lwakatare et al. (2019) conducted an empirical investigation into the performance modelling of serverless computing platforms, contrasting forecast outcomes with actual data from Amazon AWS Lambda. The study provided insights into the accuracy and applicability of performance models in predicting cost trends and resource utilisation patterns in serverless deployments.

Organisations can gain valuable insights into cost optimisation strategies, resource management practices, and performance enhancement techniques in serverless computing environments by exploring these case studies and empirical analyses of real-world serverless deployments. These studies offer practical examples of how serverless platforms can be leveraged to achieve cost-effective, scalable, and efficient solutions across various domains.

5.0. **Conclusion**

The systematic review and analysis of cost-saving mechanisms, challenges, and best practices in serverless computing have provided valuable perspectives on the complexities and opportunities of managing costs effectively in dynamic computing environments. The review has highlighted the importance of understanding cost dynamics in serverless computing for achieving optimal cost efficiency and operational effectiveness. Understanding the cost dynamics of serverless computing is crucial for practitioners and researchers to navigate the evolving landscape of cloud technologies. The potential impact of emerging trends and innovations, such as dynamic cost management tools, predictive analytics, and adaptive resource management strategies, can revolutionise how organisations approach cost optimisation in serverless environments.

Based on the study outcomes, practitioners and researchers are recommended to:

1. Embrace dynamic cost management tools to gain real-time insights into cost patterns and resource utilisation.

2. Incorporate predictive analytics and machine learning algorithms to manage costs and optimise resource allocation proactively.

3. Implement adaptive resource management strategies and workload optimisation techniques to enhance operational efficiency and foster innovation in serverless deployments.

By following these suggestions and staying current on new developments and trends in serverless cost management, organisations can enhance their cost optimisation strategies, drive innovation, and achieve sustainable cost efficiencies in their serverless deployments.

By following this structured approach, the paper can comprehensively analyse cost-saving mechanisms, challenges, and best practices in serverless computing, contributing valuable insights to the research community.

5.1 **Future Directions, Emerging Trends and Innovations in Serverless Cost Management**

Recent advancements in serverless cost management tools and methodologies are reshaping the landscape of cost optimisation in serverless computing environments. These innovations transform how organisations handle resources, manage expenses, and improve operational efficiency in serverless deployments.

One notable trend is the emergence of dynamic cost management tools that provide real-time insights into cost patterns and resource utilisation. Lei et al. (2021) highlight the significance of transformational leadership in promoting frugal innovation through knowledge sharing, underscoring the potential of dynamic cost management tools to facilitate cost-effective innovation strategies. Organisations can refine cost optimisation practices by exchanging tacit and explicit knowledge and drive frugal innovation in serverless environments.

Another key trend involves the integration of predictive analytics and machine learning algorithms in serverless cost management methodologies. Kumari et al. (2023) explore the role of workflow-aware analytical models in predicting the performance and cost of serverless executions, demonstrating how predictive analytics can enhance resource allocation and cost efficiency. By leveraging predictive models, organisations can proactively manage costs, pinpoint optimisation opportunities, and improve cost-effectiveness in serverless computing environments.

Moreover, the uptake of adaptive resource management strategies and workload optimisation techniques is gaining momentum in serverless cost management. Gallego-García et al. (2022) introduce the concept of a Dynamic Innovation Information System (DIIS) for overseeing innovation developments, emphasising adaptability and efficiency in resource allocation. By incorporating adaptive resource management practices, organisations can optimise resource usage, streamline operations, and foster innovation in serverless deployments.

These emerging trends and innovations in serverless cost management tools and methodologies can potentially transform how organisations approach cost optimisation in dynamic computing environments. By embracing dynamic cost management tools, predictive analytics, and adaptive resource management strategies, organisations can navigate the intricacies of serverless platforms, drive innovation, and achieve sustainable cost efficiencies in their deployments.

References

[1] Alsheyadi, A., Muyldermans, L., & Kauppi, K. (2019). The complementarity of green supply chain management practices and the impact on environmental performance. Journal of Environmental Management, 242, 186-198. https://doi.org/10.1016/j.jenvman.2019.04.078

[2] Aytekin, A. and Johansson, M. (2019). Exploiting serverless runtimes for large-scale optimization..

[3] https://doi.org/10.1109/cloud.2019.00090

[4] Bakken, S., Cashen, M., Mendonça, E., O'Brien, A., & Zieniewicz, J. (2000). Representing nursing activities within a concept-oriented terminological system: evaluation of a type definition. Journal of the American Medical Informatics Association, 7(1), 81-90. https://doi.org/10.1136/jamia.2000.0070081

[5] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., … & Suter, P. (2017). Serverless computing: current trends and open problems., 1-20. https://doi.org/10.1007/978-981-10-5026-8_1

[6] Benedetti, P., Femminella, M., Reali, G., & Steenhaut, K. (2021). Experimental analysis of the application of serverless computing to iot platforms. Sensors, 21(3), 928. https://doi.org/10.3390/s21030928

[7] Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. Communications of the Acm, 62(12), 44-54.

[8] https://doi.org/10.1145/3368454

[9] Das, A., Imai, S., Patterson, S., & Wittie, M. (2020). Performance optimization for edge-cloud serverless platforms via dynamic task placement..

[10] https://doi.org/10.1109/ccgrid49817.2020.00-89

[11] Denninnart, C. (2023). Efficiency in the serverless cloud paradigm: a survey on the reusing and approximation aspects. Software Practice and Experience, 53(10), 1853-1886. https://doi.org/10.1002/spe.3233

[12] Ebrahimpour, H., Ashtiani, M., & Bakhtiariazad, G. (2022). A heuristic-based package-aware function scheduling approach for creating a trade-off between cold-start time and cost in faas computing environments.. https://doi.org/10.21203/rs.3.rs-1725753/v1

[13] Ebrahimpour, H., Ashtiani, M., Bakhshi, F., & Bakhtiariazad, G. (2023). A heuristic-based package-aware function scheduling approach for creating a trade-off between cold start time and cost in faas computing environments. The Journal of Supercomputing. https://doi.org/10.1007/s11227-023-05128-z

[14] Eismann, S., Scheuner, J., Eyk, E., Schwinger, M., Grohmann, J., Herbst, N., … & Iosup, A. (2021). Serverless applications: why, when, and how?. Ieee Software, 38(1), 32-39. https://doi.org/10.1109/ms.2020.3023302

[15] Eyk, E., Scheuner, J., Eismann, S., Abad, C., & Iosup, A. (2020). Beyond microbenchmarks: the spec-rg vision for a comprehensive serverless benchmark.. https://doi.org/10.1145/3375555.3384381

[16] Gougeon, A., Lemercier, F., Blavette, A., & Orgerie, A. (2022). Modeling the end-to-end energy consumption of a nation-wide smart metering infrastructure.. https://doi.org/10.1109/iscc55528.2022.9912949

[17] Ho, T., Le, L., Thai, D., & Taewijit, S. (2016). Data-driven approach to detect and predict adverse drug reactions. Current Pharmaceutical Design, 22(23), 3498-3526. https://doi.org/10.2174/1381612822666160509125047

[18] Kelly, D., Glavin, F., & Barrett, E. (2020). Serverless computing: behind the scenes of major platforms.. https://doi.org/10.1109/cloud49709.2020.00050

[19] Lee, S., Yoon, D., Yeo, S., & Oh, S. (2021). Mitigating cold start problem in serverless computing with function fusion. Sensors, 21(24), 8416. https://doi.org/10.3390/s21248416

[20] Mampage, A., Karunasekera, S., & Buyya, R. (2022). A holistic view on resource management in serverless computing environments: taxonomy and future directions. Acm Computing Surveys, 54(11s), 1-36. https://doi.org/10.1145/3510412

[21] Mahmoudi, N. and Khazaei, H. (2022). Performance modeling of serverless computing platforms. Ieee Transactions on Cloud Computing, 10(4), 2834-2847. https://doi.org/10.1109/tcc.2020.3033373

[22] Mahmoudi and Khazaei "MLProxy: SLA-Aware Reverse Proxy for Machine Learning Inference Serving on Serverless Computing Platforms" (2022) https://doi:10.48550/arxiv.2202.11243

[23] Mohanty, S., Premsankar, G., & Francesco, M. (2018). An evaluation of open source serverless computing frameworks.. https://doi.org/10.1109/cloudcom2018.2018.00033

[24] Moher D, Liberati A, Tetzlaf J, Altman DG (2009) Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. J Clin Epidemiol 62(10):1006–1012. https://doi.org/10.1016/j.jclinepi.2009.06.005

[25] Moreno-Vozmediano, R., Huedo, E., Montero, R., & Llorente, I. (2022). Latency and resource consumption analysis for serverless edge analytics.. https://doi.org/10.21203/rs.3.rs-1457500/v1

[26] Palma, S., Garriga, M., Nucci, D., Tamburri, D., & Heuvel, W. (2021). Devops and quality management in serverless computing: the radon approach., 155-160. https://doi.org/10.1007/978-3-030-71906-7_13

[27] Rajput, K., Kulkarni, C., Cho, B., Wang, W., & Kim, I. (2022). Edgefaasbench: benchmarking edge devices using serverless computing.. https://doi.org/10.1109/edge55608.2022.00024

[28] Shafiei, H., Khonsari, A., & Mousavi, P. (2022). Serverless computing: a survey of opportunities, challenges, and applications. Acm Computing Surveys, 54(11s), 1-32. https://doi.org/10.1145/3510611

[29] Stigler, M., & Stigler, M. (2018). Understanding serverless computing. *Beginning serverless computing: developing with Amazon web services, Microsoft Azure, and Google Cloud*, 1-14. https://doi.org/10.1007/978-1-4842-3084-8_1

[30] Ustiugov, D., Petrov, P., Kogias, M., Bugnion, E., & Grot, B. (2021). Benchmarking, analysis, and optimization of serverless function snapshots.. https://doi.org/10.1145/3445814.3446714

[31] Wei, X., Lu, F., Wang, T., Gu, J., Ma, X., Chen, R., … & Chen, H. (2022). No provisioned concurrency: fast rdma-codesigned remote fork for serverless computing.. https://doi.org/10.48550/arxiv.2203.10225

[32] Wu, S., Tao, Z., Fan, H., Huang, Z., Zhang, X., Jin, H., … & Cao, C. (2021). Container lifecycle-aware scheduling for serverless computing. Software Practice and Experience, 52(2), 337-352. https://doi.org/10.1002/spe.3016

[33] Yu, T., Li, Q., Du, D., Xia, Y., Zang, B., Lu, Z., … & Chen, H. (2020). Characterizing serverless platforms with serverlessbench.. https://doi.org/10.1145/3419111.3421280