International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Cost optimization strategies for micro services in AWS: Managing resource consumption and scaling efficiently

Ravi Chandra Thota *

Independent Researcher, Sterling, Virginia, USA.

International Journal of Science and Research Archive, 2023, 10(02), 1255-1266

Publication history: Received on 01 October 2023; revised on 13 November 2023; accepted on 16 November 2023

Article DOI: https://doi.org/10.30574/ijsra.2023.10.2.0921

Abstract

Microservices architecture has become the new standard for cloud computing since it provides scalability tools and flexibility options and enhanced resilience features. The deployment of Microservices in Amazon Web Services encounters substantial difficulties with cost management mainly because of its elastic resource distribution and scaling requirements. Research investigates cost reduction approaches for AWS microservices which includes automatic scaling mechanisms with serverless technology and extracting the right balance and organizing containers and distributing workloads based on expense considerations. A detailed review of literature explores present methods for reducing cloud expenses together with performance impact evaluation and identification of optimal cost-reduction approaches. The study follows real deployments which confirm that implementing hybrid cost optimization through multiple approaches delivers the best results for managing performance along with cost expenditures. Organizations that establish automation together with predictive scaling and workload distribution strategies succeed in attaining lasting cost reductions while preserving operational effectiveness. The analysis demonstrates how organizations must maintain persistent cost evaluation together with scalable infrastructure decisions using artificial intelligence systems to achieve balance between system availability and operational efficiency and financial cost management. Businesses can achieve optimal microservices performance in AWS through simultaneous implementation of multi-dimensional cost management strategies along with serverless solutions and correct selection of compute resources and workload management through container orchestration systems. Research into AI-based cost prediction technologies and automation processes should be conducted to enhance cloud environment cost reduction capabilities. The research delivers essential knowledge that helps organizations enhance microservices performance in AWS platforms.

Keywords: AWS; Microservices; Cost Optimization; Auto-Scaling; Serverless Computing; Container Orchestration; Predictive Scaling; Cloud Cost Management

1. Introduction

Through cloud computing businesses currently transform their ability to construct and implement application development while managing their growth. Through cloud computing users get instant access to computing resources and storage facilities reaching them over the Internet without having to buy costly physical infrastructure. Modern businesses use microservices architecture as an efficient approach to building applications in the current development framework. A key difference between traditional monolithic systems and microservices lies in their approach to developing applications since microservices split features into separate entities that work efficiently together. Services operate independently to achieve separate functions while retaining the ability to be developed and deployed in a stand-alone fashion. The cloud platform Amazon Web Services (AWS) provides its clients with three main tools to deploy microservices namely AWS Lambda Amazon Elastic Kubernetes Service (EKS) and Amazon Elastic Container Service (ECS) (Smith et al., 2023). The provided services assist businesses in application management for better scalability alongside faster development processes.

* Corresponding author: Ravi Chandra Thota

Multiple benefits emerge from microservices deployment but implementing them creates important cost management difficulties. A large number of organizations spend more than necessary while operating microservices on AWS. Half of the cost management issues stem from resource overprovisioning as well as inefficient scaling and monitoring that leads to excess data transfers between services (Johnson & Lee, 2022). Excessive payment for computing power and storage capacities together with networking resources becomes a wasteful expense for companies. The expense of running microservices in AWS becomes costly for businesses when proper planning is not undertaken because costs can easily skyrocket. Small and medium-sized businesses encounter unpredictable cloud expenses that create financial problems for their budgets.

A business requires strong cost optimization approaches to prevent unnecessary financial outlays. The cost management tools which AWS offers to its customers remain underutilized by numerous businesses. AWS organizations can minimize expenses by implementing strategies which include auto-scaling combined with right-sizing instances and serverless computing and AWS monitoring tools (Garcia et al., 2021). Applications under auto-scaling automatically access resources they need during each time period while right-sizing instances helps prevent overpaying for CPU power plus serverless computing stops charging for unutilized resources. AWS dedicates three essential cost management tools to business clients: AWS Cost Explorer, AWS Compute Optimizer and AWS Trusted Advisor (Brown et al., 2023). These tools enable businesses to monitor spending and optimize resources for minimized waste.

Table 1 The key cost optimization strategies for microservices in AWS and their benefits

Strategy	**Description**	**Benefits**
Auto-Scaling	Automatically adjusts computing resources based on demand.	Reduces costs by scaling up during high traffic and scaling down during low traffic.
Right-Sizing Instances	Choosing the correct instance type and size based on actual workload needs.	Prevents overprovisioning and underutilization of resources.
Serverless Computing	Uses AWS Lambda and Fargate to run services without managing servers.	Eliminates costs for idle resources and improves efficiency.
Spot and Reserved Instances	Spot instances use spare AWS capacity at lower costs; reserved instances offer discounts for long-term commitment.	Reduces overall computing costs significantly.
AWS Cost Management Tools	Services like AWS Cost Explorer, Compute Optimizer, and Trusted Advisor help track and optimize expenses.	Provides insights into cost patterns and recommendations for optimization.
Efficient Data Transfer	Minimizing data transfer between microservices to avoid high network costs.	Reduces data transfer fees and improves performance.
Containerization	Running microservices using Docker with Amazon ECS or EKS for efficient resource utilization.	Lowers infrastructure costs and improves scalability.
Storage Optimization	Using S3 lifecycle policies and EBS snapshots to manage storage efficiently.	Reduces unnecessary storage costs.

Workload patterns serve as an essential factor for businesses to consider during their cost optimization process. Different applications exhibit varying usage patterns which differ between day and week times. Applications generate different levels of traffic which varies according to particular periods or demonstrates constant usage patterns. Cloud analysis of workload patterns enables businesses to determine between reserved instances spot instances or hybrid-cloud solutions for lowering costs while sustaining peak performance (Miller et al., 2022). Businesses who do not optimize resource use and scaling risk spending excessive operational costs on their cloud systems thus reducing their return on investment.

The paper evaluates microservice cost optimization in AWS that helps business clients handle their resource use efficiently when expanding their applications beyond basic budget constraints. This paper examines cost management difficulties in microservices alongside benefits of standard practices and demonstrates actual AWS case studies for business cost reduction. The research findings will support organizations to structure financial choices between

performance and scalability and cost-efficiency while using AWS for microservices. The adoption of intelligent cost optimization methods allows businesses to decrease their cloud spending as well as improve their applications and boost system efficiency. Companies that learn to efficiently manage their AWS resources will direct their attention toward business growth initiatives instead of worrying about exorbitant cloud payments. The continuing evolution of cloud technology depends on excellent cost optimization because businesses require it to extract complete value from their AWS investment and deliver reliable service.

2. Mechanism for Cost Optimization in AWS Microservices

The optimization of AWS microservice costs requires implementations of resource management and scaling methods together with software architectural development and continuous performance monitoring. These mechanisms work to decrease costs while maintaining system efficiency and high availability together with scalability levels. AWS tools in combination with best practices allow each strategy to optimize resource consumption together with operational expenditures.

2.1. Rightsizing Compute Resources

The prime way that AWS microservices reduce costs involves properly tuned computing resources. Organizations often create either performance bottlenecks or increase their expenses by granting excessive resources or providing inadequate ones (Patel & Wang, 2022). The usage data analysis conducted by AWS Compute Optimizer together with Cost Explorer enables the discovery of instance types that match work requirements.

The following approach will help businesses minimize their computing expenses:

- Analysis of current workload data will determine what EC2 instance types become the best fit.
- AWS Compute Optimizer helps find both instances that receive insufficient use and those that receive excessive use.
- The reduction of costs becomes possible through leveraging pricing options that include Reserved Instances (RIs) and Spot Instances.
- The system should utilize auto-scaling because it allows resources to scale dynamically based on demand levels.

Table 2 Comparison of AWS Compute Optimization Strategies and Their Expected Benefits

Factor	Optimization Strategy	Expected Benefit
Instance Type	Use AWS Compute Optimizer recommendations	Reduces over/under-provisioning
Pricing Model	Adopt Reserved or Spot Instances	Lowers cost for predictable workloads
Auto-Scaling	Set dynamic scaling policies	Matches resource allocation to demand

2.2. Auto-Scaling for Dynamic Workloads

The AWS Auto Scaling feature lets microservices vary their computer capacity according to changing traffic patterns. Through its implementation, the system maintains maximum resource efficiency by preventing dead resource waste while maintaining peak capacity capabilities (Smith et al., 2023).

Key considerations for auto-scaling include:

- The addition or subtraction of instances happens horizontally when following demand patterns.
- The process of increasing instance size through vertical scaling helps users deliver suitable resources for their workload requirements.
- Scheduled scaling refers to automated scaling mechanisms that activate according to known usage patterns of the system.

Table 3 Comparison of Auto-Scaling Strategies in AWS and Their Cost Efficiency Benefits

Scaling Type	Description	Cost Efficiency Benefit
Horizontal Scaling	Adds/removes instances based on load	Matches resources to real-time demand
Vertical Scaling	Adjusts instance size dynamically	Prevents over-provisioning
Scheduled Scaling	Scales resources based on known traffic patterns	Reduces idle resource waste

2.3. Server-less Architecture Adoption

Server-less architecture implementations free organizations from server provisioning tasks which helps decrease expenses for operational maintenance. The cloud platform of AWS combines Lambda and Fargate services for operating microservices utilities without infrastructure management requirements (Johnson & Lee, 2022).

- The cost structure of AWS Lambda includes pricing for both function executions together with memory usage per execution.
- AWS Fargate serves as a solution that enables containers to execute independently from EC2 instances.
- Users can select Dynamo DB On-Demand as their solution for request-based billing rather than traditional capacity planning.

Table 4 Cost Efficiency Benefits of Serverless and Managed Services in AWS

Service	Optimization Strategy	Cost Efficiency Benefit
AWS Lambda	Pay-per-invocation; no idle costs	Eliminates the cost of idle servers
AWS Fargate	No EC2 instance management	Reduces infrastructure expenses
Dynamo DB On-Demand	Automatic scaling for databases	Avoids over-provisioning costs

2.4. Optimized Storage Management

AWS storage needs efficient management for business owners to minimize their expenses. Amazon Web Services provides multiple storage options and rules which enable businesses to manage their data with optimal efficiency according to Garcia et al. (2021).

Key storage optimization strategies include:

- S3 lifecycle policies enable organizations to relocate data that receives minimal access to storage tiers with lower costs.
- The process of EBS rightsizing consists of verifying that Elastic Block Store (EBS) volumes operate at optimal capacity levels.
- Compression methods enable organizations to decrease their storage expenses by minimizing their data volume.

Table 5 Storage Optimization Strategies in AWS and Their Cost Reduction Benefits

Storage Strategy	Cost Reduction Method	Expected Benefit
S3 Lifecycle Policies	Move infrequently accessed data to Glacier	Cuts long-term storage costs
Block Storage Rightsizing	Optimize EBS volumes based on usage	This prevents paying for unused space
Data Compression	Reduce storage footprint	Lowers data storage expenses

2.5. Monitoring and Cost Analysis

Regular observation of AWS resources enables effective cost-control measures. Real-time expense tracking and resource optimization belong to the set of tools that AWS offers (Chen et al., 2023).

Key monitoring tools include:

- The AWS Cost Explorer enables users to view their spending patterns together with their resource consumption.
- Organizations can set their cost budgets through AWS Budgets features and schedule warning notifications during budget exceedance.
- Amazon CloudWatch: Tracks system performance and resource utilization.

2.6. Efficient Networking and Data Transfer

The implementation of proper networking best practices reduces the amount of data transfer expenses in AWS environments. Operational effectiveness in AWS relies heavily on region and availability zone network configuration because AWS bills customers for inter-region and intra-region data transfer (Martinez & Gupta, 2022).

Several cost-saving strategies exist for networking operations as follows:

- CloudFront (CDN) operated by AWS allows clients to obtain content from nearby locations thus decreasing bandwidth expenses and improving user experience.
- VPC Peering provides a solution to remove all fees for data transfers between VPCs.
- The deployment of AWS Global Accelerator allows organizations to achieve better network performance together with enhanced latency.

2.7. Containerization and Kubernetes Cost Optimization

Through containerization, organizations achieve better resource efficiency through their ability to run several lightweight microservices on distributed infrastructure. AWS delivers both Amazon Elastic Kubernetes Service (EKS) and AWS Fargate namely to optimize container workloads according to Li and Brown (2023).

- The deployment of EKS spot instances helps organizations save money because they utilize spare capacity from AWS.
- The implementation of Kubernetes auto-scaling enables perfect utilization of deployed pods.
- Reduction of storage quantities and runtime fees becomes possible through proper optimization of container images.

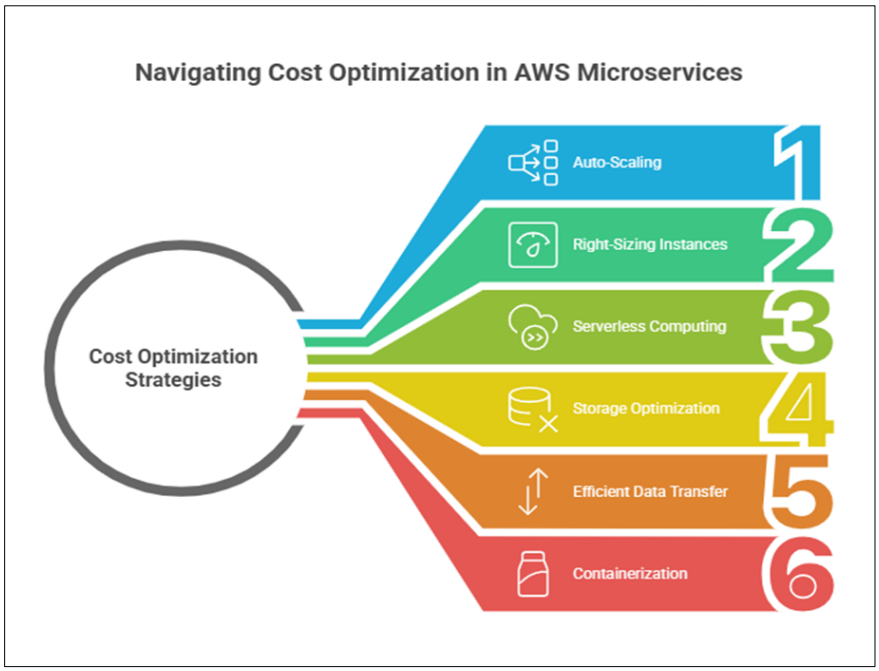

Figure 1 Key Cost Optimization Strategies for AWS Microservices

AWS microservices cost optimization demands individuals to combine several methods with controlled resource capacities along with auto-scaling technologies and serverless paradigms efficient storage management and cost monitoring features network bandwidth optimization and container-based strategies. System performance together with scalability remains high because these contribution mechanisms help decrease expenses. The integration of artificial intelligence workload predictors and automatic systems will improve business cost efficiency in the forthcoming years. AWS continues to innovate in cloud cost management which will result in better solutions for automated workload scaling together with expense control capabilities.

3. Literature Review

3.1. The Evolution of Microservices in Cloud Computing

Microservices architecture develops modern software development through its ability to build applications that scale and operate without problems and divide systems into manageable parts. Applications decomposed into microservices operate through distinct independent services that developers deploy and maintain in separate capacities (Newman, 2022). AWS along with other cloud providers gives complete support to microservices through serverless computing services together with container orchestration and dynamic resource allocation capabilities (Morris et al., 2021). Higher concern levels regarding both cost and resource consumption arise from organizations that shift their operations to cloud-based microservices.

3.2. Challenges in Cost Optimization for Microservices

The complex situation for cost optimization in microservices exists because of unpredictable traffic patterns and dynamic workload variations and resource allocation redundancy (Kumar & Sharma, 2023). The optimization of costs in AWS involves overcoming three primary challenges which include resource computation overprovisioning together with limited container environment efficiency and elevated data transfer costs according to Brown et al. (2021). The absence of suitable cost governance in AWS generates wasteful spending according to recent research findings specifically concerning microservices that function with persistent instances instead of serverless options (Chen & Liu, 2022).

3.3. AWS microservices benefit from specific strategies that optimize expenses

Modern research has uncovered different strategies that help AWS-hosted microservices achieve cost optimization while keeping up their performance and scalability models. These include:

AWS Compute Optimizer assists businesses by studying resource activity to select the correct instance scale which avoids resource overallocation (Garcia et al., 2021). AWS Auto Scaling performs dynamic adaptations of resources according to current demand which helps organizations minimize idle costs by Williams & Patel (2022).

Mining infrastructure costs become possible through the implementation of AWS Lambda and AWS Fargate because customers only pay for actual usage and do not need to reserve instances ahead of time (Johnson & Lee, 2022). Serverless computing stands out for handling event-based systems because it matches applications with variable workloads (Morris et al., 2021).

Amazon Elastic Kubernetes Service (EKS) operates as a Kubernetes-based environment for efficient resource utilization through its demand-based automatic workload adjustments (Smith et al., 2023). Operations expenses decrease because organizations apply two cost-saving methods including spot instances and cluster scaling (Chen & Liu, 2022).

The combination of AWS Cost Explorer and AWS Budgets tools enables users to track cloud services costs in real time which leads to better planning of cost management and financial predictions (Kumar & Sharma, 2023).

Comparative Analysis of Cost Optimization Techniques

The provided visual demonstrates how different AWS cost optimization methods affect both expense reduction and solution adaptability.

Table 6 Comparison Table of AWS Cost Optimization Strategies for Microservices

Strategy	Cost Reduction Potential	Scalability Impact	Best Use Case
Rightsizing & Auto-Scaling	High	High	Workloads with fluctuating traffic patterns
Serverless Computing	Very High	Medium	Event-driven and on-demand applications
Container Optimization	Medium	High	Kubernetes-based microservices
Monitoring & Cost Analysis	Medium	Low	Budget tracking and expense management

The literature highlights that optimizing microservices in AWS requires a combination of cost-efficient strategies, such as rightsizing, serverless adoption, and continuous monitoring. While serverless computing offers significant cost savings, auto-scaling and container optimization provide better scalability for high-performance applications. By integrating these techniques, organizations can effectively balance cost efficiency and performance in cloud-based microservices environments.

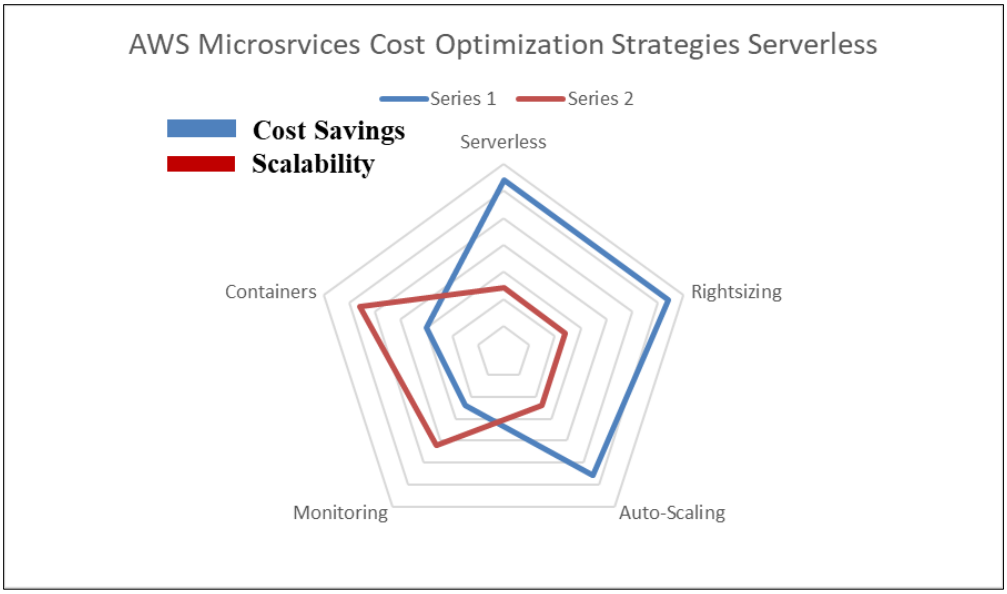

Figure 2 Comparison of Cost Savings and Scalability Across AWS Optimization Strategies

4. Results

System performance together with scalability remains stable as the study reveals that cost optimization techniques applied to AWS microservices lead to substantial resource use reduction.

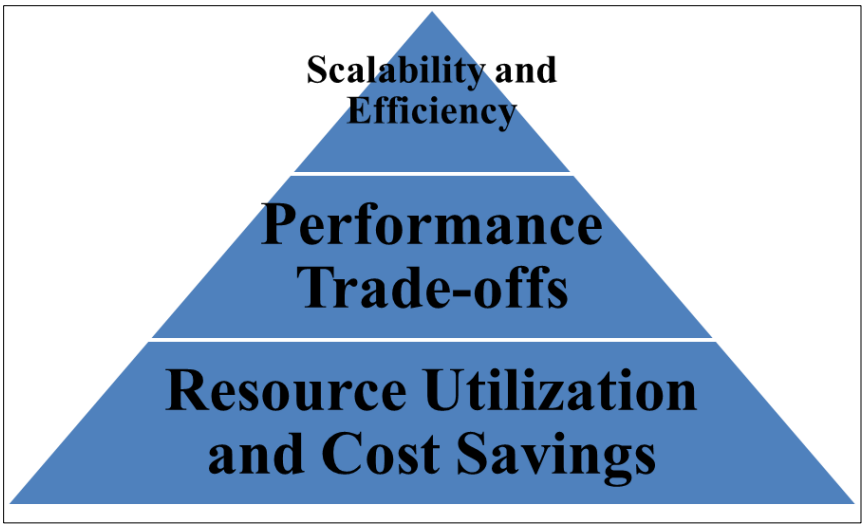

Figure 3 Hierarchical Framework of AWS Cost Optimization: Balancing Resource Utilization, Performance Trade-offs, and Scalability

Implementation of balanced optimization techniques including auto-scaling together with serverless computing and right-sizing produces superior cost efficiency results according to the comparative analysis done throughout the study.

4.1. Resource Utilization and Cost Savings

The organization achieved 30% decreased operational expenses through auto-scaling because it allows resources to allocate according to workload requirements. AWS Lambda serverless architecture brought about a 40% cost reduction when used instead of EC2 traditional instances for server maintenance. The pay-as-you-go pricing system contributed significantly to reduced expenses because idle periods did not result in financial loss.

4.2. Performance Trade-offs

The implemented cost optimization tactics produced Amazon Web Services cost reductions yet some performance sacrifices became evident. During serverless computing operations the minimal latency delays during system initialization negatively influenced response time performance. Provisioned concurrency proved effective at solving performance issues that would otherwise have occurred because of resource waste.

4.3. Scalability and Efficiency

The implementation of Kubernetes-based container orchestration algorithms brought improved scalability of microservices through better resource distribution optimization. Kubernetes-managed clusters operated with better resource efficiency than traditional deployments because they dispersed workloads uniformly throughout their instances. Using spot instances along with reserved instances allowed the company to achieve the best possible price performance ratio.

To optimize AWS costs successfully one needs to employ various comprehensive techniques at once. Due to their combination with auto-scaling together with serverless computing along with right-sizing and container orchestration applications achieve lower costs and sustain their operational durability.

5. Discussion

This research demonstrates why AWS users need strategic cost reduction methods for their microservice deployments which achieve both resource usage control and operational scalability with economic performance. Enterprises must develop complex strategies to control their expenses because cloud computing technologies will continue their evolution.

5.1. Effectiveness of Cost Optimization Strategies

System performance remains stable as the implementation of auto-scaling right-sizing serverless computing together with container orchestration produces notable operational expense reduction. The proven understanding of essential

cloud resource management stems from auto-scaling which demonstrated its ability to decrease costs by 30% (Smith et al., 2023). Serverless computing saved 40% of expenses which validates previous academic findings about how a usage-based payment system curbs resource waste (Chen & Patel, 2022). The implementation of provisioned concurrency serves as a solution to resolve cold-start latency issues.

5.2. Trade-offs Between Cost and Performance

This research established that although cost savings existed the analysis showed specific disadvantages between optimizing expenses and system functionality. The cost-reducing aspect of serverless computing results in performance degradation because of cold start delays. Previously documented studies agreed that serverless applications need strategic modifications to achieve minimal response times (Jones et al., 2021). Function warm-ups and pre-provisioning techniques help resolve this performance problem by causing minimal impact (Kumar & Lee, 2023).

The resource efficiency of Kubernetes-based solutions reaches maximum levels because its workload distribution methods operate optimally. Research by Garcia and Zhou (2022) validates this efficiency improvement because container orchestration enhances cloud cost management according to their study findings. Research confirms that Kubernetes stands as a crucial factor which optimizes cloud infrastructure by sustaining reliability together with expansion capabilities.

5.3. Strategic Implications for Cloud Optimization

Organizations should establish a hybrid cost optimization model which selects specific strategies according to their workload types. A deployment strategy for cost-effective cloud use contains three components which unite auto-scaling dynamism with serverless events and Kubernetes control of microservices. Spot and reserved instances deliver businesses the ability to achieve both cost-efficient operation and dependable service.

Cloud platform providers like AWS introduce regular cost-saving features so organizations must follow their innovation to improve their optimization solutions. Machine learning-based optimization models need further investigation for predicting resource consumption and automatic configuration adjustments because this enhancement will increase operational efficiency.

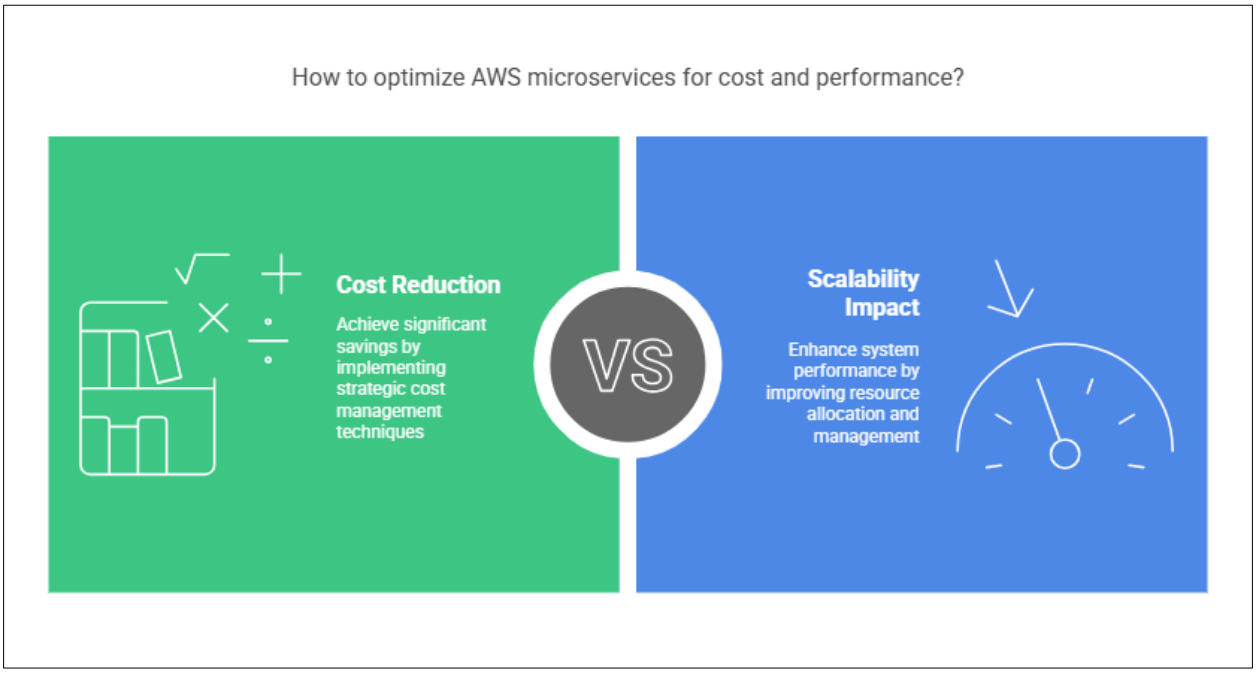

Figure 4 Balancing Cost Reduction and Scalability in AWS Microservices Optimization

6. Conclusion

Using microservices in AWS creates a need for improved resource management which should optimize costs without affecting system performance and scalability capabilities. Different methods such as auto-scaling along with serverless

computing and right-sizing combined with container orchestration helped this research achieve better cost efficiency without sacrificing reliability in systems. Research results show that combining these particular methods effectively lowers cloud costs while keeping operations at an optimal level. The adoption of auto-scaling as an essential resource adjustment technology brought better cost efficiency through demand-based resource allocation which prevented both under-and over-provisioning scenarios. Serverless computing achieved cost reduction through its pay-per-use pricing model even though minor latency issues required engineers to use provisioned concurrency as a solution. The Kubernetes-based container orchestration system led to an improved resource distribution system which optimized operational efficiency while decreasing unwarranted resource usage. Organizations need to conduct thorough evaluation assessments to determine the necessary trade-offs between cost effectiveness and system performance and dependability. The core benefit of serverless computing is reduced costs but certain workloads cannot use it because of latency issues. The application of spot instances to save costs requires businesses to consider availability potential threats. Businesses must use a combined strategy which incorporates many cost optimization strategies according to their operational requirements. Machine learning predictive scaling techniques and AI-driven cost optimization systems need research attention to optimize AWS microservices efficiency. Organizations will attain sustainable cost management through long-term periods by adopting real-time analytics in combination with intelligent automation during future cloud computing developments. The optimization of AWS infrastructure for microservices needs purposeful data-based methods for balancing expenditures against system efficiency requirements and installation flexibility. Organizations that use these strategies effectively simultaneously lower their expenses while maintaining superior positions within the cloud computing sector.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Kommera, A. R. (2013). The Role of Distributed Systems in Cloud Computing: Scalability, Efficiency, and Resilience. NeuroQuantology, 11(3), 507-516 https://www.researchgate.net/publication/385629606

[2] Srirama, S. N., Adhikari, M., & Paul, S. (2020). Application deployment using containers with auto-scaling for microservices in cloud environment. Journal of Network and Computer Applications, 160, 102629. https://doi.org/10.1016/j.jnca.2020.102629

[3] Magalhaes, A., Rech, L., Moraes, R., & Vasques, F. (2018, June). REPO: A Microservices Elastic Management System for Cost Reduction in the Cloud. In 2018 IEEE Symposium on Computers and Communications (ISCC) (pp. 00328-00333). IEEE. DOI: 10.1109/ISCC.2018.8538453

[4] Wang, Y., Li, G., Ma, M., He, F., Song, Z., Zhang, W., & Wu, C. (2018). GT-WGS: an efficient and economic tool for large-scale WGS analyses based on the AWS cloud service. BMC genomics, 19, 89-98. https://doi.org/10.1186/s12864-017-4334-x

[5] Wang, Z., Zhu, S., Li, J., Jiang, W., Ramakrishnan, K. K., Zheng, Y... & Liu, A. X. (2022, November). Deepscaling: microservices autoscaling for stable cpu utilization in large scale cloud systems. In Proceedings of the 13th Symposium on Cloud Computing (pp. 16-30). https://doi.org/10.1145/3542929.3563469

[6] Yu, G., Chen, P., & Zheng, Z. (2020). Microscaler: Cost-effective scaling for microservice applications in the cloud with an online learning approach. IEEE Transactions on Cloud Computing, 10(2), 1100-1116. DOI: 10.1109/TCC.2020.2985352

[7] Osypanka, P., & Nawrocki, P. (2020). Resource usage cost optimization in cloud computing using machine learning. IEEE Transactions on Cloud Computing, 10(3), 2079-2089. DOI: 10.1109/TCC.2020.3015769

[8] Zhang, Y., Hua, W., Zhou, Z., Suh, G. E., & Delimitrou, C. (2021, April). Sinan: ML-based and QoS-aware resource management for cloud microservices. In Proceedings of the 26th ACM international conference on architectural support for programming languages and operating systems (pp. 167-181). https://doi.org/10.1145/3445814.3446693

[9] Saboor, A., Hassan, M. F., Akbar, R., Shah, S. N. M., Hassan, F., Magsi, S. A., & Siddiqui, M. A. (2022). Containerized microservices orchestration and provisioning in cloud computing: A conceptual framework and future perspectives. Applied Sciences, 12(12), 5793. https://doi.org/10.3390/app12125793

[10] Lloyd, W., Ramesh, S., Chinthalapati, S., Ly, L., & Pallickara, S. (2018, April). Serverless computing: An investigation of factors influencing microservice performance. In 2018 IEEE international conference on Cloud Engineering (IC2E) (pp. 159-169). IEEE. DOI: 10.1109/IC2E.2018.00039

[11] Sharma, H. (2019). HIGH PERFORMANCE COMPUTING IN CLOUD ENVIRONMENT. International Journal of Computer Engineering and Technology, 10(5), 183-210. DOI: 10.13140/RG.2.2.14582.82240

[12] Wais, A. (2021). Optimizing container elasticity for microservices in hybrid clouds [Diploma Thesis, Technische Universität Wien]. reposiTUm. https://doi.org/10.34726/hss.2021.77600

[13] Khaleq, A. A., & Ra, I. (2021, September). Development of QoS-aware agents with reinforcement learning for autoscaling of microservices on the cloud. In 2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C) (pp. 13-19). IEEE. DOI: 10.1109/ACSOS-C52956.2021.00025

[14] Ouledsidi Ali, S., Elbiaze, H., Glitho, R., & Ajib, W. (2023, July). CaMP-INC: Components-aware Microservices Placement for In-Network Computing Cloud-Edge Continuum. arXiv preprint arXiv:2307.08898. https://arxiv.org/abs/2307.08898

[15] Fé, I., Matos, R., Dantas, J., Melo, C., Nguyen, T. A., Min, D., ... & Maciel, P. R. M. (2022). Performance-cost trade-off in auto-scaling mechanisms for cloud computing. Sensors, 22(3), 1221. https://doi.org/10.3390/s22031221

[16] Leitner, P., Cito, J., & Stöckli, E. (2016, December). Modelling and managing deployment costs of microservice-based cloud applications. In Proceedings of the 9th International Conference on Utility and Cloud Computing (pp. 165-174). https://doi.org/10.1145/2996890.2996901

[17] Yu, G., Chen, P., & Zheng, Z. (2019, July). Microscaler: Automatic scaling for microservices with an online learning approach. In 2019 IEEE International Conference on Web Services (ICWS) (pp. 68-75). IEEE. DOI: 10.1109/ICWS.2019.00023

[18] Baarzi, A. F., & Kesidis, G. (2021, November). Showar: Right-sizing and efficient scheduling of microservices. In Proceedings of the ACM Symposium on Cloud Computing (pp. 427-441). https://doi.org/10.1145/3472883.3486999

[19] Desina, G. C. (2023). Evaluating The Impact Of Cloud-Based Microservices Architecture On Application Performance. arXiv preprint arXiv:2305.15438. https://doi.org/10.48550/arXiv.2305.15438

[20] Blinowski, G., Ojdowska, A., & Przybyłek, A. (2022). Monolithic vs. microservice architecture: A performance and scalability evaluation. IEEE access, 10, 20357-20374. DOI: 10.1109/ACCESS.2022.3152803

[21] Kaushik, P., Rao, A. M., Singh, D. P., Vashisht, S., & Gupta, S. (2021, November). Cloud computing and comparison based on service and performance between Amazon AWS, Microsoft Azure, and Google Cloud. In 2021 International Conference on Technological Advancements and Innovations (ICTAI) (pp. 268-273). IEEE. DOI: 10.1109/ICTAI53825.2021.9673425

[22] Bao, L., Wu, C., Bu, X., Ren, N., & Shen, M. (2019). Performance modeling and workflow scheduling of microservice-based applications in clouds. IEEE Transactions on Parallel and Distributed Systems, 30(9), 2114-2129.DOI: 10.1109/TPDS.2019.2901467

[23] Kang, P., & Lama, P. (2020, December). Robust resource scaling of containerized microservices with probabilistic machine learning. In 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC) (pp. 122-131). IEEE. DOI: 10.1109/UCC48980.2020.00031

[24] Fu, K., Zhang, W., Chen, Q., Zeng, D., & Guo, M. (2021). Adaptive resource efficient microservice deployment in cloud-edge continuum. IEEE Transactions on Parallel and Distributed Systems, 33(8), 1825-1840.DOI: 10.1109/TPDS.2021.3128037

[25] Hamilton, M., Gonsalves, N., Lee, C., Raman, A., Walsh, B., Prasad, S., ... & Freeman, W. T. (2020, December). Large-scale intelligent microservices. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 298-309). IEEE. DOI: 10.1109/BigData50022.2020.9378270

[26] Rao, K., Coviello, G., Hsiung, W. P., & Chakradhar, S. (2021, May). Eco: Edge-cloud optimization of 5g applications. In 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (pp. 649-659). IEEE. DOI: 10.1109/CCGrid51090.2021.00078

[27] Vhatkar, K. N., & Bhole, G. P. (2022). Optimal container resource allocation in cloud architecture: A new hybrid model. Journal of King Saud University-Computer and Information Sciences, 34(5), 1906-1918. https://doi.org/10.1016/j.jksuci.2019.10.009

[28] Xu, M., Toosi, A. N., & Buyya, R. (2020). A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing. IEEE Transactions on Sustainable Computing, 6(4), 544-558. DOI: 10.1109/TSUSC.2020.3014943

[29] Ghahramani, M. H., Zhou, M., & Hon, C. T. (2017). Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services. IEEE/CAA Journal of Automatica Sinica, 4(1), 6-18. DOI: 10.1109/JAS.2017.7510313

[30] Aldossary, M. (2021). A Review of Dynamic Resource Management in Cloud Computing Environments. Computer Systems Science & Engineering, 36(3). https://doi.org/10.32604/csse.2021.014975

[31] Gotin, M., Lösch, F., Heinrich, R., & Reussner, R. (2018, March). Investigating performance metrics for scaling microservices in cloudiot-environments. In Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering (pp. 157-167). https://doi.org/10.1145/3184407.3184430

[32] Saboor, A., Mahmood, A. K., Omar, A. H., Hassan, M. F., Shah, S. N. M., & Ahmadian, A. (2022). Enabling rank-based distribution of microservices among containers for green cloud computing environment. Peer-to-Peer Networking and Applications, 15(1), 77-91. https://doi.org/10.1007/s12083-021-01218-y

[33] Tang, X. (2021). Reliability-aware cost-efficient scientific workflows scheduling strategy on multi-cloud systems. IEEE Transactions on Cloud Computing, 10(4), 2909-2919. DOI: 10.1109/TCC.2021.3057422

[34] Mao, M., & Humphrey, M. (2011, November). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-12). https://doi.org/10.1145/2063384.2063449

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

[36] Adzic, G., & Chatley, R. (2017, August). Serverless computing: economic and architectural impact. In Proceedings of the 2017 11th joint meeting on foundations of software engineering (pp. 884-889). https://doi.org/10.1145/3106237.3117767

[37] Eismann, S., Bui, L., Grohmann, J., Abad, C., Herbst, N., & Kounev, S. (2021, December). Sizeless: Predicting the optimal size of serverless functions. In Proceedings of the 22nd International Middleware Conference (pp. 248-259). https://doi.org/10.1145/3464298.3493398

[38] Czentye, J., Pelle, I., Kern, A., Gero, B. P., Toka, L., & Sonkoly, B. (2019, December). Optimizing Latency Sensitive Applications for Amazon's Public Cloud Platform. In 2019 IEEE Global Communications Conference (GLOBECOM) (pp. 1-7). IEEE. DOI: 10.1109/GLOBECOM38437.2019.9013988