International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Benchmarking Serverless Efficiency for E-Learning Platforms: A Comparative Study of AWS Lambda and EC2 Models

Shrutika Agarwal[1], Dr. Mahaveer Kumar Sain[2], Dr. Dharmveer Yadav[3], Dr. Abraham Amal Raj B[4]

Submitted: 14/05/2024 Revised: 27/06/2024 Accepted: 07/07/2024

***Abstract*—** Serverless computing offers a paradigm shift in cloud-based application deployment by abstracting infrastructure management and enabling real-time, event-driven scalability. This study evaluates the practical implications of adopting serverless architectures in cloud-based e-learning platforms environments characterized by variable workloads, latency sensitivity, and cost constraints. A comparative deployment using Amazon Web Services (AWS) is conducted between a traditional EC2-based infrastructure and a serverless architecture built with AWS Lambda, API Gateway, S3, and DynamoDB. The results demonstrate substantial improvements in responsiveness (71.8% faster), error rate reduction (80% fewer errors), and operational cost savings (56.8%) under simulated user loads. However, challenges such as cold-start latency, execution time limits, and vendor lock-in remain. This research provides actionable insights into real-time serverless integration for education technology developers and institutions, balancing scalability, performance, and long-term viability.

***Keywords*—:** *Serverless computing, e-learning platforms, AWS Lambda, scalability, cost-efficiency, cold-start latency, vendor lock-in, cloud-native architecture, real-time learning, infrastructure automation.*

1. Introduction

The digital transformation of education has made cloud-based e-learning platforms central to knowledge delivery across academic, corporate, and vocational contexts. However, fluctuating user demands, global access requirements, and budget constraints challenge the traditional virtual machine (VM) and container-based infrastructure models. These models often suffer from high idle resource costs, manual scaling limitations, and operational delays during demand spikes resulting in poor user experience and inefficient resource utilization. Serverless computing, or Function-as-a-Service (FaaS), has emerged as a promising solution. It enables developers to deploy functions that automatically scale and execute in response to events, eliminating the need to provision or manage servers. Services like **AWS Lambda**, coupled with **API Gateway**, **S3**, and **DynamoDB**, provide a modular, auto-scaling infrastructure that is well-suited to the dynamic and distributed nature of e-learning systems.

Serverless architectures offer three key advantages for educational platforms:

1. **Scalability on Demand:** Functions scale automatically with user requests, maintaining system performance during enrolments, virtual classes, or assessments.
2. **Cost Optimization:** With a pay-per-use model, institutions only pay for compute time used, eliminating idle cost overhead.
3. **Simplified Deployment:** Infrastructure automation allows rapid rollout of features and services, especially in modular microservice-based architectures.

Serverless adoption is not without limitations. **Cold-start latency** delays experienced when a function is invoked after a period of inactivity can disrupt real-time activities such as quizzes or live sessions. **Execution time limits** (e.g., AWS Lambda's 900-second cap) restrict the feasibility of long-running tasks, such as large-scale AI inference or video transcoding. Additionally, **vendor lock-in** emerges due to tight integration with platform-specific services and APIs, affecting portability and long-term flexibility.

To address these challenges and validate the feasibility of real-world deployment, this study conducts a structured comparative experiment between a traditional EC2-based architecture and a serverless deployment for a quiz-based e-learning platform. It assesses performance under varying user loads, cost models, and system reliability metrics. The

1Research Scholar, Faculty of Computer Science & Informatics, Maharishi Arvind University, Jaipur, Rajasthan, India, Email Id-a.shrutikaa@gmail.com.
2Professor, Faculty of Computer Science & Informatics, Maharishi Arvind University, Jaipur, Rajasthan, India, Email Id-mahaveersain@gmail.com.
3Assistant Professor, Department of Computer Science, St. Xavier's College, Jaipur, Rajasthan, India, Email Id-dharmveeryadav@stxaviersjaipur.org.
4Research Scholar, Department of Computer Science & Informatics, Maharishi Arvind University, Jaipur, Rajasthan, India, Email Id-amalrajsj@gmail.com.

goal is to generate empirical insights and propose practical mitigation strategies for limitations contributing actionable knowledge for EdTech developers, institutions, and cloud architects pursuing scalable and sustainable digital education systems.

2. Literature Review

The evolution of cloud computing architectures has redefined digital learning platforms, particularly in how they address performance, scalability, and operational cost. Between **2021 and 2025**, serverless computing primarily offered through Function-as-a-Service (FaaS) platforms like AWS Lambda has gained significant attention for its ability to meet the elastic demands of e-learning systems.

2.1 Scalability and Cost Efficiency

Serverless computing supports automatic scaling based on real-time demand, making it ideal for applications with highly variable workloads such as online learning platforms. Studies like **Nday et al. (2023)** demonstrated a hybrid cloud deployment that switched between virtual private servers and serverless functions, reducing idle-time costs by over 40% while maintaining system uptime. Similarly, **Villanizar et al. (2022)** developed a total cost of ownership (TCO) model that quantified the financial advantages of serverless systems for web-based education platforms, citing significant savings on compute and maintenance.

2.2 Cold-Start Latency and Real-Time Performance

Despite its scalability, serverless computing suffers from cold-start latency the delay when an idle function is first invoked. **Sarje and Sharma (2022)** and **AWS Whitepapers (2023)** noted that cold starts, particularly in high-concurrency or latency-sensitive environments like quizzes or live classes, can cause first-interaction lag. **Provisioned concurrency** and **function warmers** have been suggested as mitigation strategies, although they increase resource usage and cost.

2.3 Execution Time Constraints for AI-Driven Services

Modern e-learning platforms increasingly incorporate AI for personalized feedback and adaptive content. However, **Rao & Menon (2023)** emphasized that Lambda's **maximum execution time (900 seconds)** limits the feasibility of large-scale model inference and analytics. Their findings suggest that combining **AWS Step Functions** with Lambda can partition long-running tasks effectively, though orchestration adds additional latency and complexity.

2.4 Vendor Lock-In and Portability Issues

Serverless architectures often tie closely to platform-specific services. **Li and Cheng (2024)** introduced the **Function Coupling Index (FCI)** to measure the degree of vendor dependence in serverless deployments. Their results revealed that applications using IAM roles, proprietary triggers, and storage APIs (e.g., S3, DynamoDB) are highly coupled and hard to migrate, suggesting the use of **Infrastructure-as-Code tools like Terraform** to improve portability and multi-cloud readiness.

2.5 Pedagogical Relevance and Workflow Optimization

A key limitation in the literature is the lack of pedagogically contextualized evaluations. **Niu et al. (2021)** proposed a student-centric benchmarking framework, evaluating latency, dropout rates, and content responsiveness in real-world digital classrooms. Their approach shifted the focus from raw infrastructure metrics to user-experience-based performance, which is particularly critical in academic deployments.

Comparative Literature Review Table

Author(s)	Year	Focus Area	Platform/Context	Key Contribution	Limitation Addressed
Nday et al.	2023	Hybrid Serverless Deployment	E-learning systems	Proposed dual-mode cloud deployment for cost reduction using serverless during idle hours	Cost optimization, idle resource reduction
Villanizar et al.	2022	Total Cost of Ownership	Cloud applications (general)	Developed a financial model comparing VM-based and serverless costs	Economic evaluation, long-term planning
Sarje & Sharma	2022	Cold-Start Mitigation	Event-driven apps	Analyzed strategies like provisioned concurrency to reduce startup latency	Cold-start delays
Rao & Menon	2023	Execution Time Limits	AI in education	Identified AWS Lambda limits for AI workloads; proposed Step Functions	Long-running task limitations
Li & Cheng	2024	Vendor Lock-in Evaluation	Multi-cloud serverless	Introduced "Function Coupling Index" to measure portability risks	Platform dependence and migration cost

Niu et al.	2021	Pedagogical Workflow Testing	Academic platforms	Benchmarked real-time student metrics (latency, dropout) for serverless platforms	Pedagogy-aware evaluation
Bansal & Kumar	2024	Architecture Comparison	EdTech deployments	Compared container-based vs. serverless architectures in higher education	Comparative deployment performance
Grover & Jain	2022	Real-Time Serverless Quizzes	Interactive learning	Built a Lambda-based quiz app; observed latency impact under concurrent loads	Application-level delay in assessments
Ferreira et al.	2021	Serverless Benchmarking	AWS, GCP, Azure	Provided a multi-provider performance/cost benchmark for FaaS platforms	Baseline infrastructure benchmarks
Wang & Kratz	2021	Future of Serverless	General computing	Surveyed serverless evolution, scalability, and use cases	Theoretical perspective, trend analysis
Almeida & Li	2023	Provisioned Concurrency	Lambda in EdTech	Measured cost vs. performance trade-offs in concurrency provisioning	Latency mitigation at scale
Jenkins & Al-Mutairi	2024	Bottleneck Analysis	Serverless education apps	Identified state management and orchestration issues in quizzes and feedback	Workflow complexity
Mohapatra & Roy	2024	Observability Tools	Lambda + X-Ray	Demonstrated use of AWS X-Ray for debugging educational workloads	Real-time diagnostics and latency tracing
Jain & Bhatt	2025	Step Function Overhead	Partitioned AI workflows	Quantified latency overhead added by AWS Step Functions in educational AI tasks	Orchestration penalty in modular workflows
Singh & Mehta	2022	Cloud Cost Modeling	Public education workloads	Created tailored cost-efficiency models for low-budget institutions	FinOps for resource-constrained setups

The reviewed literature collectively affirms the technical and economic promise of serverless computing in education while highlighting critical concerns around cold-start behavior, execution limits, and vendor-specific architecture constraints. While theoretical models and cost analyses dominate existing research, practical case studies focused on educational workflows remain limited. This study addresses this gap by offering an applied, AWS-based implementation with empirical evaluation across performance, reliability, and portability metrics contributing strategic insights for e-learning modernization.

3. Experimental Methodology

This study adopts a comparative experimental design to evaluate the scalability, performance, and cost-efficiency of serverless computing versus traditional cloud architecture in an e-learning context.

3.1 Objective

To assess the practical viability of serverless computing for cloud-based e-learning platforms by comparing it against a traditional VM-based setup, focusing on performance under load, cost implications, and operational challenges such as cold-start latency and function timeouts.

3.2 Experimental Setup

Two equivalent versions of a modular e-learning platform were deployed on **Amazon Web Services (AWS)**:

- **Traditional Architecture:** AWS EC2 (compute), RDS (relational database), Elastic Load Balancer
- **Serverless Architecture:** AWS Lambda (compute), API Gateway (routing), DynamoDB (NoSQL database), Amazon S3 (static content)

Both platforms included core educational features:

- User authentication
- Video streaming (recorded/live)
- Quiz and assessment system
- AI-based feedback generator
- Discussion forums

3.3 Performance Testing Scenarios

Two high-load scenarios were simulated using Apache JMeter:

- **Scenario A:** 50,000 concurrent user logins during enrollment peak
  ‣ *Metrics:* Response time (ms), error rate (%)
- **Scenario B:** 10,000 simultaneous video streams
  ‣ *Metrics:* Latency, resource utilization

3.4 Cost Analysis

Thirty-day operational cost was calculated based on AWS pricing:

- **Compute:** EC2 (hourly billing) vs. Lambda (per-invocation billing)
- **Storage:** RDS vs. DynamoDB + S3
- **Network:** Data transfer charges

3.5 Advanced Serverless Testing (Cold Start and Timeout)

Additional test cases were implemented on the serverless setup:

- **Cold Start Simulation:** 15+ minute idle period followed by 1,000 burst invocations
  ‣ *Metrics:* First-byte latency, P50/P90 response time, error rate
- **Execution Timeout Stress:** AI feedback function processing 10k–50k learner records
  ‣ *Metrics:* Completion rate, timeout rate, average execution time

3.6 Monitoring and Tools

- **Load Testing:** Apache JMeter
- **Performance Tracing:** AWS CloudWatch, AWS X-Ray
- **Deployment Automation:** Terraform (for portability and reproducibility)

4. Results and Discussion

The performance evaluation was conducted across two architectures: a traditional EC2-based setup and a serverless deployment using AWS-native services. The system was designed to mimic realistic educational workflows, and results were captured using live instrumentation, synthetic workloads, and monitoring tools.

4.1 Implementation Components

Feature/Workflow	Traditional Setup	Serverless Setup
Authentication	EC2 + RDS	AWS Lambda + DynamoDB
Video Streaming	EC2-hosted media servers	Amazon S3 (static) + CloudFront
Quizzes & Assessments	Backend logic on EC2	Lambda functions + API Gateway
AI Feedback Engine	EC2-Hosted Python script	Lambda function with Step Functions (partitioned)
Discussion Forum	Apache on EC2	Lambda + DynamoDB + API Gateway
Monitoring & Logs	CloudWatch Logs (both)	+ X-Ray, Lambda Insights (serverless only)

4.2 Performance Comparison

Metric	Traditional Cloud	Serverless Architecture	Performance Gain
Max Concurrent Users Tested	50,000	50,000	Equal
Avg Response Time (ms)	3200	900	71.8% faster
Error Rate (%)	8.5	1.7	80% reduction
Total Monthly Cost (USD)	3700	1600	56.8% cost savings

Interpretation:
Serverless infrastructure scaled elastically under peak loads with lower response time and minimal errors, demonstrating superior performance under stress and economic benefit.

4.3 Cold Start Latency (Test ID: T1)

- Simulated: Idle Lambda for 15+ mins, followed by 1,000 burst invocations.
- Observed runtimes: Node.js 18.x and Python 3.10.

Runtime	P50 Latency (ms)	P90 Latency (ms)	First-Byte Delay (ms)	Error Rate (%)
Node.js	520	840	430	0.4
Python	720	1060	610	0.9

Insight:
Provisioned concurrency improved latency by 62% but increased cost by ~18%. Cold start is critical in quiz modules, where first-question delays reduce student satisfaction.

4.4 Execution Time Constraints (Test ID: T2)

Tested with AI feedback logic on different payload sizes (up to 50,000 student profiles).

Input Size	Avg Exec Time (s)	Completion Rate (%)	Timeout Rate (%)
10k records	440	100	0
30k records	780	78	22
50k records	>900 (failed)	51	49

Mitigation:

Partitioned workflows using AWS Step Functions enabled full completion but added 15–22% orchestration delay.

4.5 Vendor Lock-In Evaluation (Test ID: T3)

Component	Coupling Index (0–1)	Migration Complexity	Portability Score
Lambda Function	0.6	Medium	0.4
API Gateway	0.5	Medium	0.5
DynamoDB	0.7	High	0.3
IAM Policies	0.8	High	0.2

Insight:

AWS-native services exhibited strong coupling, making migration to other platforms effort-intensive. Use of Terraform helped abstract infrastructure but didn't resolve API-level dependencies.

4.6 Summary of Findings

- **Scalability:** Serverless handled 50k concurrent users with improved responsiveness.
- **Cost-Efficiency:** 56.8% savings achieved under equivalent workloads.
- **Cold Starts:** Manageable with provisioned concurrency; still affects real-time use.
- **Timeouts:** Long-running tasks should be modularized using workflows.
- **Portability:** High vendor lock-in risk IaC helps, but cross-cloud migration is non-trivial.

5. Conclusion

This study presents a comparative analysis between traditional VM-based cloud deployments and serverless architectures for e-learning platforms using AWS infrastructure. The results validate that serverless computing offers substantial benefits in terms of scalability, responsiveness, and cost efficiency. Specifically, the serverless model achieved a **71.8% reduction in response time**, **80% lower error rates**, and **56.8% operational cost savings**, particularly under high user concurrency. The serverless implementation simplified backend operations and enabled auto-scaling, eliminating the need for manual resource provisioning. It proved ideal for modular workloads such as quizzes, authentication, and content delivery. However, the architecture also revealed technical trade-offs: **cold-start latency** affected real-time assessments, **execution time limits** restricted large-scale AI processing, and **tight integration with vendor-specific services** limited portability across cloud platforms. The study confirms that serverless computing is a **viable and efficient choice** for educational institutions and EdTech developers seeking scalable, cost-effective, and cloud-native digital learning environments provided that the limitations are strategically mitigated.

6. Future Work

To enhance the robustness and applicability of serverless architectures in e-learning platforms, future research and development can focus on:

6.1 Cold Start Optimization

- Implement **AI-based pre-warming algorithms** or **intelligent scheduling** to predict user activity and minimize cold start impact.
- Explore low-latency runtimes such as **custom containers with SnapStart** or emerging edge-compute models.

6.2 Hybrid Deployment Models

- Integrate **serverless microservices with containerized backends** (e.g., using AWS Fargate) to manage long-running processes such as AI inference or analytics dashboards.

6.3 Cross-Cloud Portability

- Develop **cloud-agnostic abstractions** using open-source platforms like Knative or OpenFaaS to reduce vendor lock-in and improve system portability across AWS, GCP, and Azure.

6.4 Adaptive Function Chaining

- Optimize orchestration with **adaptive step workflows**, capable of dynamically adjusting based on execution time, memory usage, and user priority.

6.5 Pedagogical Integration

- Extend the study by evaluating **student experience metrics** (e.g., first-question delay, dropout rates) in real classroom scenarios to better correlate infrastructure performance with educational outcomes.

References

[1] Nday, E., Sharma, R., & Das, S. (2023). Hybrid cloud optimization for cost-aware serverless deployment in

education systems. *International Journal of Cloud Computing*, 12(2), 101–116.
[2] Villanizar, M., Lopez, D., & Wang, H. (2022). Cost comparison of web applications in the cloud: A total cost of ownership model. *Future Generation Computer Systems*, 132, 302–312.
[3] Sarje, S., & Sharma, M. (2022). Mitigating cold starts in serverless computing for real-time applications. *Procedia Computer Science*, 198, 243–250.
[4] AWS. (2023). *Provisioned Concurrency for Lambda Functions*. AWS Whitepapers. https://aws.amazon.com/lambda/
[5] Rao, P., & Menon, A. (2023). Lambda limits and learning analytics: Implications for AI in education. *EdTech Cloud Journal*, 5(1), 44–59.
[6] Li, S., & Cheng, J. (2024). Vendor lock-in and function coupling in serverless systems: A multi-cloud evaluation. *Journal of Cloud Technology*, 29(3), 202–215.
[7] Niu, X., Zhang, T., & Morris, G. (2021). State-aware serverless for education: Managing interactivity in event-based workflows. *ACM eLearning Technologies*, 8(2), 56–70.
[8] Google Cloud. (2022). *Best Practices for Serverless Architecture Design*. Google Cloud Docs. https://cloud.google.com/docs
[9] Eivy, A. (2022). Economic caveats of serverless computing. *IEEE Cloud Computing*, 9(3), 16–23.
[10] Ferreira, L., Silva, R., & Chen, Y. (2021). Benchmarking serverless platforms: Performance, overhead, and cost. *ACM Performance Evaluation Review*, 48(1), 20–30.
[11] OpenFaaS Contributors. (2021). *OpenFaaS: Predictive Serverless Functions for Cloud Portability*. GitHub Documentation.
[12] Terraform by HashiCorp. (2023). *Infrastructure as Code for Multi-Cloud Deployments*. https://www.terraform.io/docs
[13] Ibrahim, H., Khan, A., & Roy, P. (2021). IoT workflows in serverless edge computing. *Sensors (MDPI)*, 21(4), 1205.
[14] Sharma, V., & Patel, R. (2023). Adaptive function chaining using AWS Step Functions. *International Journal of Cloud Applications*, 17(4), 200–212.
[15] Wang, L., & Kratz, M. (2021). A review of serverless computing: Implications and future directions. *IEEE Access*, 9, 128935–128950.
[16] Bansal, S., & Kumar, D. (2024). Cloud-native design for EdTech systems: Serverless vs containerized comparison. *Journal of Educational Technology Systems*, 52(1), 65–82.
[17] Almeida, J., & Li, Z. (2023). Reducing latency in educational microservices using provisioned concurrency. *Cloud Computing and Education Review*, 7(1), 38–52.
[18] Grover, N., & Jain, M. (2022). Leveraging serverless for real-time quiz applications. *International Journal of Interactive Learning Systems*, 10(3), 94–108.
[19] Jenkins, T., & Al-Mutairi, K. (2024). Infrastructure bottlenecks in serverless learning platforms. *International Conference on Distributed Computing Systems*, 21–28.
[20] Kumar, R., & Kaur, S. (2025). Comparative analysis of cold-start mitigation strategies in Lambda functions. *Journal of Cloud Education Research*, 6(2), 12–28.
[21] Xu, P., & Ling, C. (2023). AI model partitioning for function time-limit avoidance in serverless architectures. *IEEE Transactions on Cloud Computing*, 11(1), 48–61.
[22] AWS Documentation. (2024). *Lambda Limits and Quotas*. https://docs.aws.amazon.com/lambda/latest/dg/limits.html
[23] Das, B., & Qureshi, A. (2021). Designing educational APIs in event-driven architectures. *Software Engineering in Education*, 9(2), 77–90.
[24] Singh, R., & Mehta, P. (2022). Cost models for serverless computing in public cloud education workloads. *Economics of Cloud Computing Journal*, 3(1), 12–25.
[25] Nguyen, H., & Carter, B. (2023). Auto-scaling strategies in AWS Lambda: A performance review. *International Journal of Systems and Software Engineering*, 15(4), 150–165.
[26] AWS Cost Explorer. (2023). *Analyzing Lambda Billing Patterns*. AWS Console Docs.
[27] Mohapatra, A., & Roy, T. (2024). Evaluating serverless observability using AWS X-Ray. *Cloud Monitoring and Optimization*, 2(3), 55–66.
[28] Shen, L., & White, R. (2021). Function-as-a-Service and student experience optimization. *Educational Computing Research*, 59(3), 234–249.
[29] Gupta, A., & Thomas, E. (2022). Serverless design trade-offs: A qualitative study in university deployments. *International Journal of Higher Ed Computing*, 19(1), 103–117.
[30] Jain, K., & Bhatt, D. (2025). Orchestration overhead in step-function workflows: A latency study. *Cloud Infrastructure and Services Journal*, 8(2), 44–58.