INTEGRATION OF IAAS AND SERVERLESS COMPUTING FOR E-COMMERCE ORDER PROCESSING

Verda MISIMI JONUZI[1], Miranda XHAFERI [2], Florim IDRIZI

[1]Department of Informatics, Faculty of FNSM, North Macedonia*
[2] Department of Mathematics, Faculty of FNSM, North Macedonia
[*]Corresponding Author: e-mail: verda.misimi@unite.edu.mk

Abstract

Infrastructure as a Service (IaaS) and serverless computing represent complementary paradigms within modern cloud architectures. IaaS provides persistent virtualized infrastructure for hosting applications and databases, whereas serverless computing enables event-driven execution without direct server management. This study examines their integration in e-commerce order-processing systems through a hybrid architectural approach.
The proposed model combines IaaS-based virtual machines with serverless functions responsible for transaction validation, inventory synchronization, and customer notification workflows. AWS Lambda and Azure Functions are analyzed as representative serverless platforms within commercial cloud ecosystems. The study adopts a conceptual architectural analysis methodology focused on scalability, modularity, workload responsiveness, and separation of processing logic from persistent infrastructure services.
The analysis demonstrates that integrating serverless execution with IaaS environments can improve resource utilization and operational flexibility in systems characterized by fluctuating transaction workloads. Event-driven processing reduces unnecessary infrastructure allocation while maintaining responsiveness during periods of increased activity. The study contributes a structured hybrid cloud framework that supports the development of scalable, adaptive, and cost-aware e-commerce architectures.

Keywords: Infrastructure as a Service (IaaS), serverless computing, AWS Lambda, Azure Functions, cloud computing, e-commerce.

Introduction

Infrastructure as a Service (IaaS) represents a fundamental cloud computing model that provides virtualized computing resources over the internet, including processing power, storage, and networking capabilities. These resources are hosted and managed by external providers, allowing organizations to allocate infrastructure according to operational needs without maintaining physical hardware (Minarolli, 2014).
In this model, responsibility is shared between provider and user. While the provider manages physical infrastructure, users configure and maintain operating systems, applications, and deployed services. This separation supports flexible scaling and adaptation to changing workloads. Virtualization enables multiple virtual machines to operate on shared physical resources while remaining isolated. This improves resource utilization and supports dynamic workload distribution across applications.
Serverless computing has emerged as a complementary approach within cloud architectures. It allows application functions to execute in response to events without direct server management, while resource allocation and scaling are handled automatically by the provider (Hassan et al., 2021).
Unlike traditional always-on infrastructure, serverless systems are better suited for intermittent and event-driven workloads such as data updates, user requests, and transaction processing. Services such as AWS Lambda and Azure Functions provide execution environments for such tasks within cloud ecosystems.
E-commerce systems represent a typical environment where workload variability is significant. Traffic and transaction volume fluctuate depending on user activity, requiring architectures that can scale efficiently without excessive resource provisioning (Shi & Dustdar, 2016; Li et al., 2020).

Recent research has explored the combined use of IaaS and serverless computing as a hybrid approach for improving scalability and workload distribution in cloud-based systems.
This study examines how virtualized infrastructure and event-driven serverless functions can operate together within e-commerce order-processing environments to improve scalability and operational efficiency.

1.1.Research Contribution

This study contributes to cloud architecture research in three specific ways.
First, it presents a structured hybrid model that explicitly combines Infrastructure as a Service (IaaS) with serverless computing within a unified e-commerce order-processing framework. Unlike conventional studies that treat these paradigms separately, this work focuses on their operational interaction within a single workflow.
Second, the study introduces a comparative architectural mapping between AWS Lambda and Azure Functions in relation to equivalent IaaS components. This allows a clearer understanding of how vendor-specific ecosystems influence system design decisions.
Third, the research identifies key architectural criteria—scalability behavior, modularity, workload responsiveness, and separation of concerns—as analytical dimensions for evaluating hybrid cloud systems. These criteria can be reused in future studies examining similar cloud integration patterns.

Integration of Infrastructure as a Service (IaaS) and Serverless Computing for E-Commerce Order Processing

The integration of Infrastructure as a Service (IaaS) and Serverless Computing provides a flexible approach for managing cloud-based e-commerce systems. In this model, IaaS resources are used to maintain the core infrastructure components of the platform, while serverless functions handle dynamic processing tasks that require automatic scaling and event-driven execution.
E-commerce applications typically involve multiple operations, including user authentication, product management, inventory control, payment processing, and customer notifications.
These processes generate workloads that may vary significantly depending on user activity and transaction volume. Combining IaaS and serverless technologies allows organizations to distribute these workloads more efficiently across different cloud services.
In the proposed architecture, virtual machines hosted within the IaaS environment are responsible for maintaining the main website, application services, and databases. These components require continuous availability and stable resource allocation in order to support customer access and data management operations.
At the same time, serverless computing services are used for processing customer orders and executing event-driven tasks. When a customer places an order, a serverless function is automatically triggered to validate the transaction, update inventory records, store order information, and generate notifications. This execution model enables automatic resource scaling according to demand without requiring manual server management.
AWS Lambda and Azure Functions are examined as representative serverless platforms that support this architecture. Both services provide event-driven execution, integration with cloud storage systems, and automatic scaling capabilities that are suitable for e-commerce environments.
Combining persistent infrastructure with event-driven execution can reduce unnecessary resource allocation while improving scalability in cloud-based commerce systems. In addition, separating persistent infrastructure components from dynamic processing functions allows organizations to manage workloads more efficiently while maintaining flexibility within the cloud environment.

Methodology

This study adopts a conceptual architectural analysis approach focused on hybrid cloud system design. The objective is to examine how Infrastructure as a Service (IaaS) and serverless computing can be structurally integrated within e-commerce order-processing environments.
The research design is qualitative and model-based. Instead of implementing a live system or performing performance benchmarking, the study evaluates architectural configurations, service interactions, and functional distribution across cloud layers.
Evaluation is guided by four analytical dimensions: scalability behavior, system modularity, responsiveness to workload variation, and separation between infrastructure and processing logic. These dimensions are used to assess the suitability of serverless integration within IaaS-based systems.
The analysis combines workflow modeling with technology mapping across AWS and Azure environments. This includes identifying how event-driven functions interact with virtualized infrastructure components to support core e-commerce operations such as order validation, inventory synchronization, and notification delivery.
The study focuses on design-level interpretation of hybrid cloud behavior under variable workloads, rather than empirical performance measurement.

System Architecture and Workflow Design

Based on the defined evaluation framework, the system architecture is developed as a conceptual model of hybrid cloud interaction.

The proposed model combines Infrastructure as a Service (IaaS) and Serverless Computing to support order-processing operations within an e-commerce environment. The architecture is designed to separate persistent infrastructure components from event-driven processing tasks in order to improve scalability and operational efficiency.
Within the IaaS environment, virtual machines are used to host the main e-commerce website, application services, and databases. These virtualized resources maintain continuous system availability and support customer interactions, product management, and data storage operations. The databases store product information, customer profiles, inventory records, and order history.
Serverless computing services handle event-triggered processing, reducing reliance on continuously running infrastructure and enabling dynamic resource allocation.
The workflow begins when a customer accesses the e-commerce platform through the web application hosted on the IaaS infrastructure. After selecting products and completing the checkout process, the system generates an event that activates a serverless function responsible for order processing.
The serverless function performs several operations, including:

- retrieval of order information,
- validation of transaction data,
- updating inventory records,
- storing order details in the database,
- and generating customer notifications.

After the processing stage is completed, the updated information is synchronized with the database hosted within the IaaS environment. Notification services may also be triggered to inform customers about order confirmation and delivery status.
This architecture allows the platform to respond dynamically to varying transaction volumes while minimizing the need for manual infrastructure management. In addition, the separation

between web hosting services and processing functions supports modular system design and improves scalability during periods of increased customer activity.

Azure Functions Implementation

Azure Functions was selected as one of the serverless platforms examined in this study due to its integration capabilities within Microsoft Azure cloud environments. Azure Functions supports event-driven execution and automatic scaling, enabling applications to process workloads without maintaining continuously active servers (Microsoft, 2023a).

In the proposed implementation, Azure Functions are responsible for handling order-processing tasks triggered by customer transactions. The serverless function communicates with supporting Azure services to manage data storage, inventory tracking, and notification delivery.

The following Azure services are integrated within the architecture:

- **Azure Blob Storage** is used for storing product-related information, customer data, and order records.
- **Azure Cosmos DB** maintains inventory information and supports real-time updates of order status.
- **Azure Service Bus** enables communication between services and supports customer notification delivery.

When a customer confirms a purchase, the Azure Function is automatically triggered. The function retrieves transaction details, validates the request, updates inventory records, and stores order information in the associated database services. The system then generates a notification confirming successful order processing. This implementation supports scalable execution while reducing the administrative overhead associated with traditional server management.

AWS Lambda Implementation

AWS Lambda is examined as an alternative serverless platform within the proposed architecture. AWS Lambda enables developers to execute application functions without directly managing servers or infrastructure resources. The platform automatically scales according to incoming requests and supports integration with multiple AWS cloud services (Amazon Web Services, n.d.).

Within the AWS-based implementation, order events initiate Lambda execution when customers complete transactions on the e-commerce platform. The functions process order-related operations and communicate with supporting AWS services for storage and notification management.

The supporting AWS ecosystem includes the following components:

- **Amazon S3** is used for storing customer information, product data, and order records.
- **Amazon DynamoDB** maintains inventory status and supports real-time database operations.
- **Amazon SNS (Simple Notification Service)** is used for sending order confirmation and status notifications to customers.

Once an order is submitted, the AWS Lambda function validates transaction information, updates inventory data, stores order records, and initiates notification services. Automatic scaling mechanisms allow the platform to handle varying transaction loads without requiring manual allocation of additional infrastructure resources. The AWS Lambda implementation demonstrates how serverless computing can support efficient and scalable order-processing operations within IaaS-based e-commerce systems.

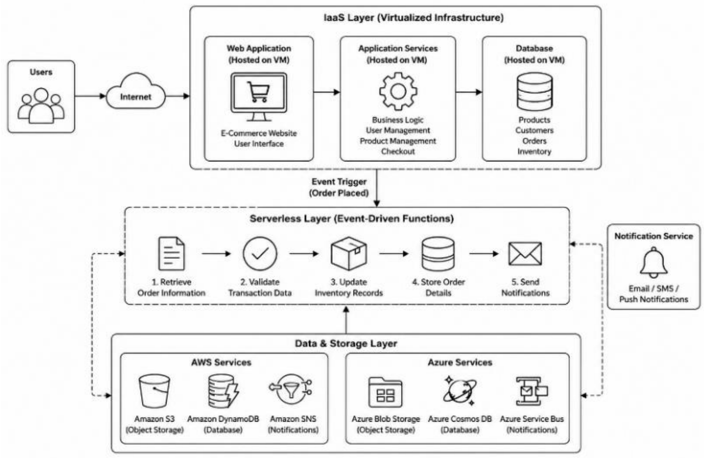

Figure 1. Proposed Hybrid IaaS and Serverless Architecture for E-Commerce Order Processing

The proposed architecture consists of three layered components that collectively support scalable e-commerce order processing.

At the **presentation layer**, users interact with a web application hosted on virtual machines within the IaaS environment. This layer handles user requests, product browsing, and checkout operations.

The **processing layer** is composed of serverless functions deployed on AWS Lambda and Azure Functions. These functions are triggered by order-related events and are responsible for transaction validation, inventory updates, and notification handling.

The **data layer** includes cloud-based storage and database services such as Amazon S3, DynamoDB, Azure Blob Storage, and Cosmos DB, which store persistent application data including product information, customer profiles, and order history.

The interaction between these layers follows an event-driven flow where user actions generate triggers that activate serverless execution, while persistent services maintain system state and data consistency.

Comparative Analysis and Discussion

AWS Lambda and Azure Functions both provide event-driven execution environments that can be integrated into IaaS-based e-commerce systems. Although they follow similar architectural principles, their practical use is often shaped by ecosystem alignment and system dependencies rather than functional differences alone.

AWS Lambda is more frequently adopted in systems built around Amazon Web Services, where integration with services such as Amazon S3, DynamoDB, and SNS enables consistent workflow design across storage, computation, and messaging components. In contrast, Azure Functions is more naturally aligned with Microsoft Azure environments, integrating with services such as Blob Storage, Cosmos DB, and Service Bus.

Scalability is a shared feature of both platforms, allowing automatic adjustment of computing resources based on workload demand. This is particularly relevant in e-commerce systems, where traffic patterns are irregular and often unpredictable.

From a cost perspective, both platforms follow usage-based billing models, where charges depend on execution time and consumed resources rather than provisioned capacity. However, actual operational cost may vary depending on execution frequency, memory configuration, and external service dependencies.

In practice, platform selection is often influenced by organizational familiarity with a specific cloud ecosystem rather than technical superiority. This reduces integration complexity and simplifies long-term maintenance. Overall, both services are suitable for implementing serverless order-processing functions in IaaS-based architectures, provided that they are aligned with the surrounding cloud infrastructure.
Table 1 presents a general comparison between AWS Lambda and Azure Functions within the context of e-commerce order processing.

Table 1. Comparison of AWS Lambda and Azure Functions

Feature	AWS Lambda	Azure Functions
Cloud Ecosystem	Amazon Web Services	Microsoft Azure
Main Storage Service	Amazon S3	Azure Blob Storage
Database Integration	DynamoDB	Cosmos DB
Notification Service	Amazon SNS	Azure Service Bus
Scaling Mechanism	Automatic	Automatic
Billing Model	Pay-per-use	Pay-per-use
Infrastructure Management	Managed by AWS	Managed by Azure
Suitable for Event-Driven Tasks	Yes	Yes

Conclusions

This study focused on the integration of Infrastructure as a Service (IaaS) and serverless computing in the context of e-commerce order-processing systems. The proposed framework combines persistent virtualized infrastructure with event-driven processing functions to support scalable cloud-based operations.
The analysis showed that IaaS is suitable for hosting core system components such as web applications and databases, while serverless computing is more efficient for handling dynamic processing tasks triggered by user activity. AWS Lambda and Azure Functions were used as representative technologies to demonstrate this integration.
The findings indicate that separating infrastructure hosting from transactional processing can improve system scalability and reduce operational complexity. In addition, serverless execution enables more efficient handling of workload variations without continuous resource allocation.
Both AWS Lambda and Azure Functions were found to be effective for implementing such workflows, with selection depending mainly on the underlying cloud ecosystem and organizational requirements.
The proposed hybrid model demonstrates that combining IaaS infrastructure with serverless execution can support scalable and operationally flexible e-commerce environments.

Recommendations and Future Work

The integration of Infrastructure as a Service (IaaS) and Serverless Computing continues to attract attention in cloud system design due to its potential for improving scalability and operational flexibility. In practice, however, its effectiveness depends on workload characteristics and system configuration, which should be further examined in applied environments.

Additional research may also compare the execution efficiency of AWS Lambda and Azure Functions using performance indicators such as response time, scalability, execution latency, and operational cost. Experimental benchmarking in high-demand environments could provide more detailed insight into the behavior of serverless platforms during periods of increased workload.

Organizations implementing cloud-based e-commerce systems should consider monitoring and cost-management mechanisms as part of their deployment strategy. Services such as AWS CloudWatch and Azure Cost Management may assist in tracking resource utilization and identifying operational inefficiencies.

Security and access control remain important considerations in cloud architectures. Future implementations should incorporate authentication policies, identity management mechanisms, and data protection strategies in order to support secure transaction processing and compliance with organizational requirements.

Another area for future investigation involves the development of vendor-independent architectures capable of operating across multiple cloud providers. Such approaches may improve portability and reduce dependency on a single cloud ecosystem.

Finally, future studies may examine the applicability of combining IaaS and serverless computing in domains beyond e-commerce, including healthcare systems, logistics platforms, and financial applications that require scalable and event-driven cloud infrastructure.

References

[1]. Akkus, I. E., Chen, R., Rimac, I., Stein, M., Satzke, K., Beck, A., Aditya, P., & Hilt, V. (2018). *SAND: Towards high-performance serverless computing. In **Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 2018)** (pp. 923–935). USENIX Association.*

[2]. Amazon Web Services. (n.d.). *What is AWS Lambda?* https://aws.amazon.com/lambda/

[3]. Amazon Web Services. (2023). *AWS CloudWatch documentation*. https://docs.aws.amazon.com/cloudwatch/

[4]. Armbrust, M., et al. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50–58.

[5]. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., & Mitra, T. (2017). Serverless computing: Current trends and open problems. *Research Advances in Cloud Computing*.

[6]. Eivy, A. (2017). Be wary of the economics of "serverless" cloud computing. *IEEE Cloud Computing, 4*(2), 6–12. https://doi.org/10.1109/MCC.2017.32

[7]. Gupta, H., Dastjerdi, A. V., Ghosh, S. K., & Buyya, R. (2021). iFogSim: A toolkit for modeling and simulation of resource management techniques in IoT, edge, and fog computing environments. *Software: Practice and Experience, 47*(9), 1275–1296. https://doi.org/10.1002/spe.2509

[8]. Hassan, H. B., Barakat, S. A., & Sarhan, Q. I. (2021). Survey on serverless computing. *Journal of Cloud Computing*, 10(1), 39. https://doi.org/10.1186/s13677-021-00253-7

[9]. Hendrickson, S., Sturdevant, S., Klabnik, S., & Arpaci-Dusseau, A. (2020). Serverless is more: From PaaS to present cloud computing. *Communications of the ACM, 63*(2), 54–63. https://doi.org/10.1145/3368455

[10]. Jonas, E., et al. (2019). *Cloud programming simplified: A Berkeley view on serverless computing*. arXiv. https://arxiv.org/abs/1902.03383

[11]. Kumar, P., & Goudar, R. H. (2022). Cloud security issues and challenges: A survey. *International Journal of Computer Applications*.

[12]. Li, Z., Ouyang, W., & Lu, Y. (2020). Benchmarking performance of serverless computing platforms. *Journal of Cloud Computing, 9*(1), 1–16. https://doi.org/10.1186/s13677-020-00179-3

[13]. Lorido-Botrán, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of Grid Computing*, 12(4), 559–592. https://doi.org/10.1007/s10723-014-9314-7

[14]. Microsoft. (2023a). *Azure Functions overview*. https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview

[15]. Microsoft. (2023b). *Azure cost management and billing documentation*. https://learn.microsoft.com/en-us/azure/cost-management-billing/

[16]. Microsoft. (2023c). *Introduction to Azure Blob Storage*. https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

[17]. Minarolli, D. (2014). *Utility-based allocation of resources to virtual machines in cloud computing* (Doctoral dissertation, University of Marburg).

[18]. Shi, W., & Dustdar, S. (2016). The promise of edge computing. *Computer*, *49*(5), 78–81. https://doi.org/10.1109/MC.2016.145

[19]. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18. https://doi.org/10.1007/s13174-010-0007-6