International Journal of Scientific Research and Modern Technology (IJSRMT) ijsrmt.com
Volume 4, Issue 7, 2025
DOI: https://doi.org/10.38124/ijsrmt.v4i7.1045

From IaaS to FaaS: An Evolutionary Analysis of Cloud Service Models

Raghuvar Karthik Durga[1]

Publication Date: 2025/07/29

Abstract

Cloud computing has undergone a significant transformation through the development of service models, which are gradually shifting more infrastructure management responsibilities to cloud. This research examines the complete development of cloud computing services, spanning from IaaS to PaaS and FaaS. The paper demonstrates that this technological progression represents a strategic adaptation to meet the rising needs of developers for productivity and operational requirements for efficiency and business flexibility. The paper examines IaaS, PaaS, and FaaS through their essential features, benefits, and drawbacks, showing how they balance control with convenience and cost-effectiveness. The research presents a systematic evaluation that demonstrates how various models support specific design approaches and financial needs. The paper examines the key elements of containerization and microservices architecture that have enabled this technological advancement. The paper establishes that cloud-native development will succeed through the combined use of IaaS, PaaS, and FaaS within a unified system, allowing the developers to select the best solution for each application requirement.

Keywords: *Cloud Computing, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Function-as-a-Service (FaaS), Serverless Computing, Cloud Service Models, Evolutionary Analysis, Cloud Architecture, Operational Abstraction, Microservices.*

I. INTRODUCTION

➢ *The Paradigm Shift in Computing*

Cloud computing has revolutionized the fundamental structure of information technology through its single transformative paradigm. The traditional on-premises infrastructure model limited businesses and developers due to its capital-intensive requirements, which demanded substantial investments in physical equipment, dedicated facilities, and continuous maintenance personnel. The traditional infrastructure model operated at a slow pace, offering limited flexibility and a poor ability to adapt to changing market needs (Kratzke, 2018). Cloud computing brought about a revolutionary change from traditional asset-based infrastructure to a flexible utility-based system. The internet enabled users to access computing power and storage resources as standardized commodities, which they could instantly deploy and scale up or down while only paying for actual usage. The infrastructure transition has made top-tier resources accessible to all businesses, allowing small companies to compete with established corporations while accelerating innovation across every market sector (Kratzke, 2018).

➢ *The Cloud Service Model Spectrum requires definition*

The cloud model consists of multiple distinct abstraction layers, which we refer to as service models. The service models establish boundaries between cloud consumers and providers through their spectrum, which ranges from basic infrastructure to full application delivery. The fundamental models that demonstrate this progression include IaaS, which delivers virtualized computing resources; PaaS, which manages application development and deployment environments; SaaS, which provides complete applications; and FaaS, which represents serverless computing for event-triggered code execution. The paper examines the development from IaaS to FaaS because this sequence demonstrates the industry's advancement toward higher levels of abstraction and better efficiency. Table .1 illustrates the evolution of management responsibility from user-managed (On-Premises, IaaS) to provider-managed (PaaS, FaaS, SaaS), highlighting the core concept of increasing abstraction (Sharma, 2015).

Table 1 The Spectrum of Cloud Service Abstraction: Division of Responsibility (Devin, 2025).

IT Component	On-Premises	IaaS (e.g., AWS EC2)	PaaS (e.g., Heroku)	FaaS (e.g., AWS Lambda)	SaaS (e.g., Gmail)
Applications	User	User	User	User	Provider
Data	User	User	User	User/Provider	Provider
Runtime	User	User	Provider	Provider	Provider
Middleware	User	User	Provider	Provider	Provider
O/S	User	User	Provider	Provider	Provider
Virtualization	User	Provider	Provider	Provider	Provider
Servers	User	Provider	Provider	Provider	Provider
Storage	User	Provider	Provider	Provider	Provider
Networking	User	Provider	Provider	Provider	Provider
Legend	User-Managed	User-Managed	User-Managed	User-Managed	Provider-Managed

The paper demonstrates that software architecture underwent a complete transformation from IaaS to FaaS, as developers sought higher abstraction levels and operational efficiency, as well as business agility, rather than just different procurement methods. The evolution of software development enables developers to shift from handling virtual hardware to executing code, while the cloud provider handles operational tasks, allowing developers to focus on value creation. The research examines each model separately before evaluating its trade-offs and assessing how this transition will affect software development practices in the future.

II. THE FOUNDATION: INFRASTRUCTURE AS A SERVICE (IaaS)

Iaas represents the base level of cloud computing services, which operates as "Virtualization as a Service." The main benefit of IaaS lies in its ability to transform physical hardware components, such as servers, storage, and networking equipment, into virtualized, scalable resources that users can access through the internet (Gibson, 2012) (Sotomayor, 2009). The IaaS model enables users to access virtualized computing infrastructure over the internet, while the cloud provider operates and maintains the physical hardware in its data centers. Organizations can utilize IaaS to access IT infrastructure on a usage-based payment model, eliminating the need for capital expenditures and physical data center management (Xu, 2014).

➢ *Key Characteristics and Technologies*

The IaaS model provides users with complete control and maximum flexibility when utilizing its services. The fundamental technological elements of this model consist of:

- *Virtual Machines:*

  Virtual Machines (VMs) operate as the primary compute resources, duplicating physical computers to enable users to deploy their operating systems and applications. Amazon Elastic Compute Cloud (EC2) and Microsoft Azure Virtual Machines serve as examples of virtual machine services.

- *Virtual Storage:*

  The virtual storage system includes durable, scalable services that enable users to store data persistently through Amazon S3 object storage and Azure Disks block storage for VMs and applications.

- *Virtual Networks:*

  Users can establish isolated networks through software-defined networking to manage firewalls and direct internet traffic between their cloud resources and external networks.

Users can manage resources through a management console or APIs, which provide the same level of control as physical hardware but at a much faster pace and with greater flexibility (Feng, 2014).

➢ *Advantages*

- *Granular Control:*

  IaaS provides users with complete control over their infrastructure, which results in several key benefits. The main benefits enable them to customize the operating system, middleware, and runtime environment at a detailed level (Marshall, 2011).

- *Flexibility:*

  The system enables IaaS due to its high level of control and flexibility. It provides users with administrative access to VMs, allowing them to adjust their IT infrastructure at any time to match their changing requirements (Marshall, 2011).

- *"Lift-and-Shift" Migration Ease:*

  IaaS provides an optimal solution for cloud migration of existing on-premises applications because it allows users to transfer their applications with minimal changes through exact environment duplication (Marshall, 2011).

➢ *Limitations and Challenges*

The IaaS model provides advantages, yet it creates major operational obstacles for users.

- *High Operational Overhead:*

  The user must handle operating system management through patching and security updates and runtime environments and middleware and application maintenance. The management of IaaS requires substantial IT knowledge together with continuous maintenance work.

- *Cost Unpredictability:*
  The pay-as-you-go pricing model of IaaS becomes difficult to predict when users fail to properly control their resource usage. The ongoing payment for idle VMs results in unnecessary financial waste because they continue to generate charges.

- *Resource Underutilization:*
  Organizations end up wasting money on inefficient spending because they provide excessive resources to handle peak loads when they lack advanced scaling policies. The operational management of infrastructure now replaces capital expenses, but organizations must handle all operational complexities (Sun, 2011).

III. THE RISE OF ABSTRACTION: PLATFORM AS A SERVICE (PaaS)

Platform-as-a-Service (PaaS) was developed as a direct solution to address the significant operational challenges presented by IaaS. The solution of IaaS eliminated the need for hardware acquisition and maintenance, but developers and IT staff still needed to handle operating system management, database, middleware, and runtime environment maintenance (Kamateri, 2013). The primary purpose of PaaS development was to manage infrastructure, allowing developers to focus on application development and innovation. PaaS functions as a complete "Development Platform as a Service" solution. The platform enables developers to build web applications through its managed cloud-based environment, which supports all stages from development to testing, deployment, and application management. The primary advantage of PaaS is that cloud providers manage all infrastructure components, ranging from servers to storage, networking, operating systems, middleware, and runtime environments, allowing developers to focus on application code development and optimization (Gibson, Benefits and challenges of three cloud computing service models., 2012).

➢ *Key Characteristics*
The PaaS model contains three main defining characteristics, which are:

- *Automated Deployments:*
  The PaaS model enables automated application deployment through streamlined workflows that originate from code repositories, requiring reduced configuration.

- *Integrated Development Tools:*
  The platform includes integrated development tools that support developers in debugging and monitoring application performance and addressing development needs.

- *Managed Data Services:*
  The platform offers managed data services through Amazon Relational Database Service (RDS) and Azure SQL Database, which handle database provisioning, maintenance, and backup operations.

- *Integrated Application Services:*
  The platform provides pre-built application services for authentication, caching, and messaging, which developers can easily integrate into their applications (Kolb, 2014) (Nanda, 2024).

➢ *Advantages*
The main benefits of PaaS stem from its ability to enhance operational efficiency and developer productivity.

- *Increased Developer Productivity:*
  The platform enables developers to deploy features at a faster rate, as they are not required to handle platform management tasks (Agrawal, 2010).

- *Reduced Operational Complexity:*
  The provider assumes full responsibility for platform maintenance, encompassing security updates, system monitoring, and platform updates (Agrawal, 2010).

- *Inherent Scalability:*
  PaaS platforms include automatic scaling features that adjust resource allocation according to application usage patterns, eliminating the need for developer involvement (Agrawal, 2010).

➢ *Limitation*
The basic design of PaaS systems offers both high productivity and simplicity of operation, but it creates particular challenges for users.

- *Reduced Control:*
  The operating system, server configuration, and network settings remain inaccessible to developers due to their limited or no control over these elements, which creates restrictions for applications requiring low-level functionality (Kolb, 2014).

- *Vendor Lock-in:*
  The use of proprietary APIs, deployment workflows, and services by PaaS platforms leads to vendor lock-in situations that make platform migration both difficult and expensive (Kolb, 2014).

- *Platform-Specific Learning Curves:*
  The adoption of a PaaS platform requires developers to learn its unique tools and methodologies, resulting in both increased training time and reduced adaptability during the initial development phase (Kolb, 2014).

IV. THE PINNACLE OF ABSTRACTION: FUNCTION AS A SERVICE (FaaS) / SERVERLESS

Function-as-a-Service (FaaS) operates as serverless computing, which represents the natural progression of cloud abstraction beyond PaaS application-centricity to establish event-driven computing as its core model (Barcelona-Pons, 2019). FaaS operates as serverless computing, which breaks down applications into their most minor functional units that execute when specific events occur. The transition requires developers to shift from administering servers for extended applications to executing temporary code for particular purposes. FaaS functions as a cloud service that delivers "Event-Driven, Ephemeral Execution Environments" according to its definition (Li, 2023). The FaaS model requires developers to create simple, stateless code blocks (functions) that execute a single task at a time. The cloud provider handles all aspects of compute resource provisioning and function execution within temporary containers, then completely shuts down the environment when the function stops running. The system design requires developers to write only business logic code because the provider takes full responsibility for runtime execution (Barcelona-Pons, 2019).

➢ *Key Characteristics and Technologies*

The FaaS model operates through multiple essential technological features, which include:

- *Event Sources:*

Functions in the FaaS model receive activation through various types of events, including HTTP requests via an API Gateway, file uploads to storage buckets, database modifications, and queue message reception (Hoseinyfarahabady, 2017).

- *Statelessness:*

Functions operate without a state because they cannot maintain in-memory data between separate runs. External services, including databases and caches, must store all necessary state information to ensure seamless operation (Hoseinyfarahabady, 2017).

- *Ephemeral Containers:*

The code executes within temporary containers that exist only during execution time before they are automatically destroyed (Hoseinyfarahabady, 2017).

- *Fine-Grained Billing:*

Cloud computing utilizes a billing system that offers the most detailed pricing structure, charging users based on execution counts and millisecond computing time, rather than fixed capacity allocations (Hoseinyfarahabady, 2017).

➢ *Advantages*

FaaS provides transformative benefits for particular application scenarios.

- *Accurate Pay-Per-Use Pricing:*

The pay-per-use pricing model of FaaS enables users to pay only for the time their code runs, measured in milliseconds. The system charges no fees for idle resources, which results in substantial cost reductions for workloads that occur sporadically (Kuhlenkamp, 2019).

- *Ultimate Operational Abstraction ("NoOps"):*

The developer experiences complete abstraction from all infrastructure components through the "NoOps" model, which includes operating systems and runtimes. The provider manages all aspects of infrastructure provisioning, scaling, patching, and fault tolerance operations (Kuhlenkamp, 2019).

- *Automatic and Instantaneous Scaling:*

A FaaS platform enables automatic and immediate scaling between zero and thousands of parallel executions based on event rates without requiring any pre-configuration (Kuhlenkamp, 2019).

➢ *Limitations and Challenges*

The high degree of abstraction in this system creates multiple complex operational difficulties:

- *Cold Start Latency:*

A function invocation delay known as cold start happens when the platform needs to create a new container because the function has been inactive, which negatively affects applications that require fast responses (Yussupov, 2019).

- *Execution Duration Limits:*

The execution time limits set by cloud providers range from 5 to 15 minutes, which makes FaaS unsuitable for running extended processes (Yussupov, 2019).

- *Debugging and Monitoring Complexity:*

The process of tracing and debugging distributed applications that use numerous short-lived functions becomes more complicated than traditional monolithic applications and PaaS deployments (Yussupov, 2019).

- *Heightened Vendor Lock-in:*

The application architecture becomes so tightly integrated with the FaaS implementation, event sources, and supporting services of the provider that migration becomes extremely challenging (Yussupov, 2019).

V. COMPARATIVE ANALYSIS: A STRUCTURED EVALUATION

A comprehensive understanding of the IaaS to FaaS evolution necessitates a systematic evaluation of operational and architectural elements. The analysis goes past basic descriptions to reveal essential trade-offs that determine the best model for present-day applications (Malla, 2019).

Table 2 Comparative Analysis of Cloud Service Models (Lee, 2021) (Kolb, Towards Application Portability in Platform as a Service, 2014).

Dimension	IaaS	PaaS	FaaS
Level of Abstraction	Low (Virtualized Hardware)	Medium (Managed Platform)	Very High (Event-Driven Logic)
Operational Responsibility	Dev manages OS, Runtime, Data, & App	Cloud manages OS & Runtime; Dev manages Data & App	Cloud manages almost everything; Dev manages Function Logic
Scaling Model	Manual / Predictive (by the user)	Automated (based on app-level metrics)	Automated, Granular, Instantaneous (per function)
Billing Granularity	Per provisioned unit (e.g., vCPU/hour)	Per application instance/time	Per request (execution time & memory)
Development Agility	Low (High setup & maintenance)	High (Focus on code)	Very High (Focus purely on business logic)
Performance (Startup Time)	Minutes (VM boot time)	Seconds (Container initialization)	Milliseconds (but with cold start delays)
Vendor Lock-in Potential	Low (standard VMs are portable)	Medium (proprietary platform services)	High (proprietary triggers & environments)

➢ *Analysis of Trade-Offs*

The data in Table 2. demonstrates a consistent pattern, showing that the evolution from IaaS to FaaS requires developers to choose between complete system control, operational efficiency, and developer productivity. The IaaS model provides users with complete control over their computing environment through its operation. Users who lease land under IaaS can design and construct any building using any available materials. The complete freedom to operate comes with substantial operational expenses, extended development times, and inefficient resource management because organizations must pay for unused capacity. The scaling process operates at a slow pace through manual intervention, which makes it unsuitable for handling unpredictable workload patterns (Kolb, Towards Application Portability in Platform as a Service., 2014).

PaaS achieves an optimal position in this trade-off between control and productivity. The platform offers developers high productivity and operational simplicity through its managed environment, which requires them to relinquish direct control over the OS and runtime environment. The managed platform of PaaS enables teams to focus on application code and data while accelerating their time-to-market delivery. The automated scaling of PaaS operates with reduced precision compared to FaaS; however, developers must still pay for idle application instances. The operational efficiency and development agility of FaaS reach their peak through the complete abstraction of server concepts, resulting in a "NoOps" experience and a cost-effective pay-per-execution model (Malla, 2019). The developer no longer needs to consider servers because FaaS provides a complete "NoOps" experience for code execution, utilizing a payment system based on execution counts that reduces costs. The system provides developers with minimal control, as they must work within time constraints and create stateless applications while addressing cold start issues and vendor-specific integration challenges. The development path shows no indication of replacing one service with another because each service maintains its unique strengths. The optimal choice for workloads that need particular operating system configurations and extended operations and migration through "lift-and-shift" exists on IaaS platforms. The development and deployment of full-fledged web applications and APIs requires PaaS because it provides the highest level of developer speed. FaaS delivers its best performance in situations that need automatic scaling and cost efficiency during sporadic or event-driven operations with variable workloads. The modern cloud-native application utilizes multiple service models in combination, as each service model offers its own unique benefits (Wen, 2023).

VI. DRIVERS AND IMPLICATIONS OF EVOLUTION

The transition from IaaS to FaaS represents more than sequential technological advancements because it stems from fundamental technological and business changes that reshape software architecture and developer responsibilities (Long, 2020).

➢ *Technological Drivers*

The containerization revolution enabled the evolution through Docker and Kubernetes technologies as its primary drivers. The container technology established a lightweight, portable, and standardized method for packaging application code together with its dependencies. The serverless model requires isolated, minimal execution environments that start quickly, so FaaS needed this capability as a necessary step. The serverless model depends on container technology to deliver its core characteristics of fast scaling and short-lived execution, even though FaaS hides these components from developers. The industry's transition toward fine-grained, scalable components became evident through Kubernetes orchestration, which manages distributed, complex systems (Lin, 2023).

➢ *Business Drivers*

The main business drivers behind this transformation stem from organizations seeking continuous speed, financial efficiency, and operational agility. The digital economy requires companies to introduce new market features at a rate that outpaces their competitors to gain a competitive advantage. The transition from IaaS to FaaS enables organizations to eliminate the unnecessary work of managing infrastructure because it handles this process automatically. The financial transformation under this evolution enables businesses to move from capital-intensive hardware purchases (CapEx) to operational expenses (OpEx) that scale with usage. The pay-per-execution pricing model of FaaS represents the pinnacle of this financial transformation, as it converts fixed IT expenses into costs that directly align with business operations (Hoseinyfarahabady, A QoS-Aware Resource Allocation Controller for Function as a Service (FaaS) Platform, 2017).

➢ *Architectural Implications*

The service model transformation is in harmony with the microservices architecture. The transition from monolithic applications running on IaaS VMs to independent, deployable services became possible due to PaaS and FaaS. The requirement to handle and expand these tiny services demonstrated the importance of PaaS and FaaS management systems. The evolution of microservices architecture has reached its natural conclusion with FaaS, as each service now operates as a single function, which the cloud provider manages for deployment and scaling. The shift in software design has occurred toward event-driven message-based patterns because these patterns work optimally in serverless environments (Casale, 2019).

➢ *The Role of the Developer*

Software developers undergo a significant transformation in their work responsibilities due to the development of cloud service models. During the IaaS period, developers needed to handle system administration tasks ("sysadmin-coders") because they maintained virtual machine health and configuration. The PaaS model required developers to concentrate on application development through code and data management. The development of FaaS technology has led developers to focus entirely on business logic execution. Developers in serverless environments today function as event-driven workflow architects, utilizing cloud services to create business solutions while minimizing their time spent on infrastructure management (Sewak, 2018).

VII. CHALLENGES AND THE ROAD AHEAD

➢ *Persistent Challenges in Cloud Computing*

Cloud computing technology continues to encounter ongoing difficulties, primarily stemming from security issues, performance problems, and standardization challenges. The practice of outsourcing services to third-party providers poses significant security risks, as it compromises data privacy and hinders the ability to meet regulatory requirements (Anjana & Singh, 2018). The unpredictable nature of cloud service performance, combined with network issues, creates reliability concerns for cloud-based services (Dikaiakos et al., 2009). The absence of standardization between cloud platforms creates interoperability problems, which force users to experience difficulties when they attempt to switch providers, as it requires substantial work and results in data loss (Saraswat & Tripathi, 2020). Cloud infrastructure operators must focus on developing more effective methods to manage energy consumption and reduce power usage in their systems (Jing et al., 2011).

➢ *Is FaaS the Final Stage?*

Function as a Service (FaaS) represents a current development in cloud computing, but it does not indicate the absolute end of its evolution. The integration of serverless containers with AI-driven optimization systems indicates that cloud computing will develop into a unified system. AWS Fargate operates as a serverless container system that unites the advantages of IaaS, PaaS, and FaaS functionality. The deployment and scaling processes become simpler because serverless containers handle runtime environments and provide scalability without requiring users to manage infrastructure (Kolb & Wirtz, 2014). AI-driven autonomous optimization systems optimize resource allocation to create more efficient cloud operations, resulting in decreased operational expenses. The development of serverless technologies and AI systems will create an advanced cloud computing system, which will drive ongoing innovation and operational efficiency according to Dikaiakos et al. (2009).

➢ *The Coexistence of Models*

FaaS operates as a complementary technology to IaaS and PaaS by establishing a hybrid cloud-native system, which enables multiple models to coexist for various use cases. The infrastructure needs of legacy systems require IaaS to function correctly, as it enables users to customize their environment and maintain complete control over their infrastructure. The development and deployment of web applications becomes more efficient through PaaS because it provides integrated tools that simplify application lifecycle management. The event-driven processing capabilities of FaaS make it suitable for handling data stream processing and HTTP request responses without server provisioning requirements (Zhou et al., 2010; Bokhari et al., 2017). Modern applications implement a multi-model approach to achieve performance optimization and cost reduction by combining their most valuable features for handling different workload requirements (Genez et al., 2012).

VIII. CONCLUSION

The paper follows the significant development of cloud service models, starting with IaaS (Infrastructure as a Service), which virtualizes hardware, then moving to PaaS (Platform as a Service), which manages applications, and finally to FaaS (Function as a Service), which enables event-driven execution. The development exhibits a direct path of rising abstraction, as each stage transfers operational control from users to cloud providers. The evolution has transformed resource consumption patterns by shifting from virtual computer rentals to short-lived business logic execution (Kratzke, 2018) (Sharma, 2015). The transition from IaaS to FaaS represents an architectural and philosophical transformation that goes beyond basic procurement modifications. The primary reason behind this change is that developers seek to achieve better operational efficiency, faster market entry, and enhanced business flexibility. The primary trade-off lies between relinquishing detailed system management capabilities for enhanced developer efficiency and reduced operational expenses. The primary organizational objective is to enable businesses to allocate their resources toward innovation rather than managing infrastructure operations. The development of cloud service models represents an ongoing process that enables users to gain greater control and power (Yussupov, 2019). The core value of cloud computing exists in its ability to eliminate obstacles that prevent developers from delivering their value to users. The abstraction of hardware, platform, and server complexities through cloud computing has enabled developers to create, test, and deploy software at speeds and efficiencies that were previously unimaginable. The future of software development will emerge from the strategic combination of IaaS, PaaS, and FaaS because each model brings distinct advantages that will enable developers to create more significant software solutions at unprecedented speeds (Xu, 2014).

REFERENCES

[1]. Kratzke, N. (2018). A brief history of cloud application architecture. Applied Sciences, 8(8), 1368.

[2]. Sharma, S. (2015). Evolution of as-a-Service Era in Cloud. arXiv preprint arXiv:1507.00939.

[3]. Saraswat, M., & Tripathi, R. C. (2020). Cloud Computing: Analysis of Top 5 CSPs in SaaS, PaaS and IaaS Platforms. 300–305. https://doi.org/10.1109/smart50582.2020.9337157

[4]. Kolb, S., & Wirtz, G. (2014). Towards Application Portability in Platform as a Service. 281, 218–229. https://doi.org/10.1109/sose.2014.26

[5]. Bokhari, M. U., Tamandani, Y. K., & Makki, Q. (2017). A Survey on Cloud Computing (Vol. 1, Issue 11, pp. 149–164). Springer Singapore. https://doi.org/10.1007/978-981-10-6620-7_16

[6]. Jing, S.-Y., Ali, S., Zhong, Y., & She, K. (2011). State-of-the-art research study for green cloud computing. The Journal of Supercomputing, 65(1), 445–468. https://doi.org/10.1007/s11227-011-0722-1

[7]. Zhou, M., Qian, W., Zhang, R., & Zeng, D. (2010). Services in the Cloud Computing era: A survey. 40–46. https://doi.org/10.1109/iucs.2010.5666772

[8]. Genez, T. A. L., Madeira, E. R. M., & Bittencourt, L. F. (2012). Workflow scheduling for SaaS / PaaS cloud providers considering two SLA levels. 906–912. https://doi.org/10.1109/noms.2012.6212007

[9]. Anjana, A., & Singh, A. (2018). Security concerns and countermeasures in cloud computing: a qualitative analysis. International Journal of Information Technology, 11(4), 683–690. https://doi.org/10.1007/s41870-018-0108-1

[10]. Dikaiakos, M. D., Mehra, P., Katsaros, D., Vakali, A., & Pallis, G. (2009). Cloud Computing: Distributed Internet Computing for IT and Scientific Research. IEEE Internet Computing, 13(5), 10–13. https://doi.org/10.1109/mic.2009.103

[11]. Gibson, J., Tan, Q., Eveleigh, D., & Rondeau, R. (2012). Benefits and challenges of three cloud computing service models. 2, 198–205. https://doi.org/10.1109/cason.2012.6412402

[12]. Sotomayor, B., Llorente, I. M., Montero, R. S., & Foster, I. (2009). Virtual Infrastructure Management in Private and Hybrid Clouds. IEEE Internet Computing, 13(5), 14–22. https://doi.org/10.1109/mic.2009.119

[13]. Xu, F., Vasilakos, A. V., Liu, F., & Jin, H. (2014). Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Directions. Proceedings of the IEEE, 102(1), 11–31. https://doi.org/10.1109/jproc.2013.2287711

[14]. Feng, Y., Li, B., & Li, B. (2014). Price Competition in an Oligopoly Market with Multiple IaaS Cloud Providers. IEEE Transactions on Computers, 63(1), 59–73. https://doi.org/10.1109/tc.2013.153

[15]. Marshall, P., Freeman, T., & Keahey, K. (2011). Improving Utilization of Infrastructure Clouds. 949, 205–214. https://doi.org/10.1109/ccgrid.2011.56

[16]. Sun, A., Xiong, F., Ji, T., & Yue, Q. (2011). IaaS Public Cloud Computing Platform Scheduling Model and Optimization Analysis. International Journal of Communications, Network and System Sciences, 04(12), 803–811. https://doi.org/10.4236/ijcns.2011.432098

[17]. Kamateri, E., Ahtes, J., D'Andria, F., Lobunets, O., Gouvas, P., Loutas, N., Ravagli, F., Ledakis, G., Zeginis, D., Bocconi, S., & Tarabanis, K. A. (2013). Cloud4SOA: A Semantic-Interoperability PaaS Solution for Multi-cloud Platform Management and Portability (pp. 64–78). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40651-5_6

[18]. Kolb, S., & Wirtz, G. (2014). Towards Application Portability in Platform as a Service. 281, 218–229. https://doi.org/10.1109/sose.2014.26

[19]. Nanda, A. K., Augustine, P. J., Kiran, V., Sharma, A., Cyril, B. R., & Sampath, B. (2024). Securing Cloud Infrastructure in IaaS and PaaS Environments (pp. 1–33). Igi Global. https://doi.org/10.4018/979-8-3693-1431-9.ch001

[20]. Agrawal, D., Das, S., & El Abbadi, A. (2010). Big data and cloud computing. Proceedings of the VLDB Endowment, 3(1–2), 1647–1648. https://doi.org/10.14778/1920841.1921063

[21]. Barcelona-Pons, D., Sánchez-Artigas, M., París, G., Gómez-Gómez, A., Ruiz, Á., & García-López, P. (2019). FaaS Orchestration of Parallel

Workloads. 25–30. https://doi.org/10.1145/3366623.3368137
[22]. Li, X., Leng, X., & Chen, Y. (2023). Securing Serverless Computing: Challenges, Solutions, and Opportunities. IEEE Network, 37(2), 166–173. https://doi.org/10.1109/mnet.005.2100335
[23]. Hoseinyfarahabady, M., Lee, Y. C., Zomaya, A. Y., & Tari, Z. (2017). A QoS-Aware Resource Allocation Controller for Function as a Service (FaaS) Platform (pp. 241–255). Springer. https://doi.org/10.1007/978-3-319-69035-3_17
[24]. Kuhlenkamp, J., Borges, M. C., Werner, S., Tai, S., & El Tal, K. (2019). An Evaluation of FaaS Platforms as a Foundation for Serverless Big Data Processing. 1–9. https://doi.org/10.1145/3344341.3368796
[25]. Yussupov, V., Wurster, M., Leymann, F., & Breitenbücher, U. (2019). A Systematic Mapping Study on Engineering Function-as-a-Service Platforms and Tools. 229–240. https://doi.org/10.1145/3344341.3368803
[26]. Malla, S., & Christensen, K. (2019). HPC in the cloud: Performance comparison of function as a service (FaaS) vs infrastructure as a service (IaaS). Internet Technology Letters, 3(1), e137. https://doi.org/10.1002/itl2.137
[27]. Lee, I. (2021). Pricing and Profit Management Models for SaaS Providers and IaaS Providers. Journal of Theoretical and Applied Electronic Commerce Research, 16(4), 859–873. https://doi.org/10.3390/jtaer16040049
[28]. Wen, J., Jin, X., Liu, X., & Chen, Z. (2023). Rise of the Planet of Serverless Computing: A Systematic Review. ACM Transactions on Software Engineering and Methodology, 32(5), 1–61. https://doi.org/10.1145/3579643
[29]. Long, J., Yuan, M. J., Tai, H.-Y., & Hsieh, S.-T. (2020). A Lightweight Design for Serverless Function as a Service. IEEE Software, 38(1), 75–80. https://doi.org/10.1109/ms.2020.3028991
[30]. Lin, C., Fan, C., Khazaei, H., & Mahmoudi, N. (2023). Fine-Grained Performance and Cost Modeling and Optimization for FaaS Applications. IEEE Transactions on Parallel and Distributed Systems, 34(1), 180–194. https://doi.org/10.1109/tpds.2022.3214783
[31]. Casale, G., Jakovits, P., Long, M., Artač, M., Wurster, M., Leymann, F., Russo, A., Srirama, S. N., Papanikolaou, V., Tamburri, D. A., Van Hoorn, A., Presenza, D., Van Den Heuvel, W.-J., & Zhu, L. (2019). RADON: rational decomposition and orchestration for serverless computing. SICS Software-Intensive Cyber-Physical Systems, 35(1–2), 77–87. https://doi.org/10.1007/s00450-019-00413-w
[32]. Sewak, M., & Singh, S. (2018, April 1). Winning in the Era of Serverless Computing and Function as a Service. https://doi.org/10.1109/i2ct.2018.8529465
[33]. Devin. (2025, April 4). *IaaS, PaaS, SaaS: Which is Best for Your Business?* M5 Internet Hosting. https://www.m5hosting.com/blog/iaas-paas-saas-which-is-best-for-your-business/