The Impact of Serverless Computing on Cost Optimization

Parag Bhardwaj

Abstract

Cloud computing has undergone a revolutionary paradigm with serverless computing, it can significantly optimize costs during application development and deployment. Instead of the traditional server based model, serverless computing abstracts infrastructure management completely, leaving developers to only focusing on code execution, from provisioning and maintaining servers. Pay-as-you-go pricing model of this model charges users only for the actual compute resources consumed during execution without charging for the idle server time. Therefore, serverless computing can save companies a lot of money, especially when applied on applications with variable or unpredictable workloads. This allows businesses to scale resources automatically based on demand, which means that resources can be used efficiently and that overprovisioning or underutilization of resources, which is often the case with traditional server based environments, do not exist. High levels of elasticity can be achieved through Serverless platforms like AWS Lambda, Azure Functions and Google Cloud Functions which allows organizations to scale up during peak loads and not have to pay for additional resources in off peak periods. In addition, serverless computing reduces operational overhead as it does not require system administration tasks such as patching, scaling and infrastructure management. While serverless computing has a lot to gain from serverless computing regarding potential for cost savings, managing complex workflows or maintaining stateful applications may present challenges. However, if used correctly, it can result in faster, more agile and cheaper application development that can create higher value for businesses.

Introduction

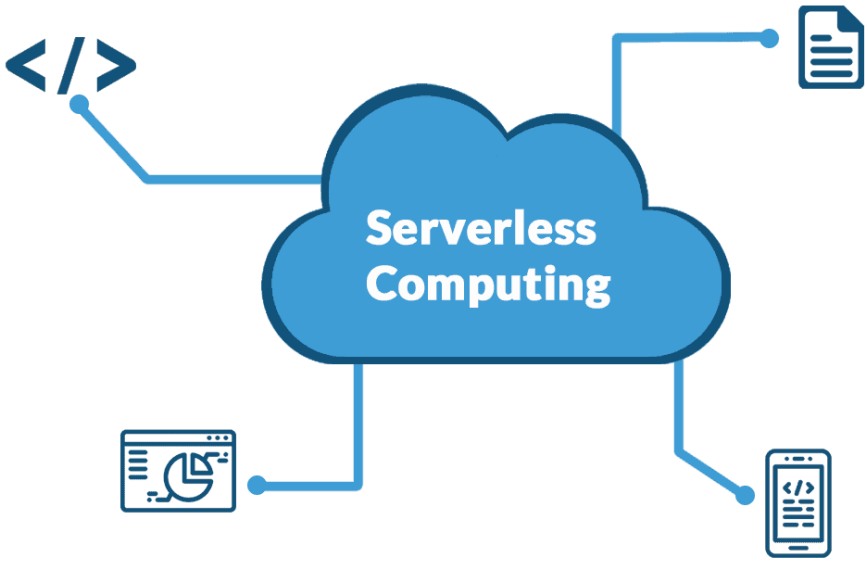
Serverless computing has taken hold in cloud computing because of its ability to reduce costs and simplified application development. In the past, businesses would need to provision and manage servers to run their applications and this meant high capital expenditure: on hardware, on maintenance, and scaling. Serverless computing removes the need for you to manage servers at all because the cloud provider takes care of it. This allows developers to write and deploy code to run on demand, without concern over those soft servers and resources underneath. Businesses benefit from this approach because they can avoid making up front hardware investments and minimal operational overhead of server maintenance, software patching, and scaling. AWS Lambda, Azure Functions, and Google Cloud Functions are serverless platforms that run on pay per use basis —a user only has to pay for the resources they actually use, depending on the execution time and the resources they consumed rather than the number or size of servers needed. Therefore, organizations can realize considerable cost savings by paying only for what they use; instead of having to provision over resources or spend time on idle server time.

Other than cost savings, serverless (even cloud standings) computing also brings in scalability and efficiency which makes the cost further optimized. In traditional server based models, like a business provisioning resources for peak load, businesses are left with underutilized resources during off peak hours. However, serverless computing automatically scales according to the number of requests it receives, so only as many resources will be used to provide it as it is needed, minus the resources automatically allocated by the system to keep things running smoothly. This elasticity balances the risk of overpaying for unused

capacity as well as guaranteeing that the application will always remain highly available during periods of high demand. Additionally, serverless computing ends up saving the day as you don't have to manage the system anymore and developers can simply create and deploy the application. With businesses migrating to serverless architectures they can become more agile, reduce cost and improve overall operational operational efficiency, making serverless a more popular option in modern application development.

Overview of Serverless Computing

Cloud native execution model that enables cloud providers to automatically handle infrastructure, while developers simply write and deploy code with no provision or management of servers required. Serverless does not mean that there is no server but rather that there is no need to manage the server, providing it, scaling it, maintaining it, and so on, because all of that is run by the cloud provider. What are serverless platforms, such as AWS Lambda, Azure Functions, or Google Cloud Functions for that matter, that allow developers to run functions or small code segments in response to API requests, file uploads, or database updates without the need for manual intervention in server management? Unlike traditional hosting models, serverless computing is a pay per use model, where customers pay for real computation time the applications consume rather than large dedicated server capacity regardless of how much used.



The scalability, flexibility and cost optimization aspect make this a very beneficial model. Serverless platforms let the application resources automatically scale up or down based on demand, so the application only uses as much processor power as necessary, preventing over usage (and thus over provisioning). Such dynamic scaling is precisely what is needed for applications with unbalanced workloads or dynamic traffic. Serverless computing also negates the costs that businesses have to put in the necessary infrastructure or dedicated servers as resources are tapped on demand. Furthermore, abstraction of infrastructure

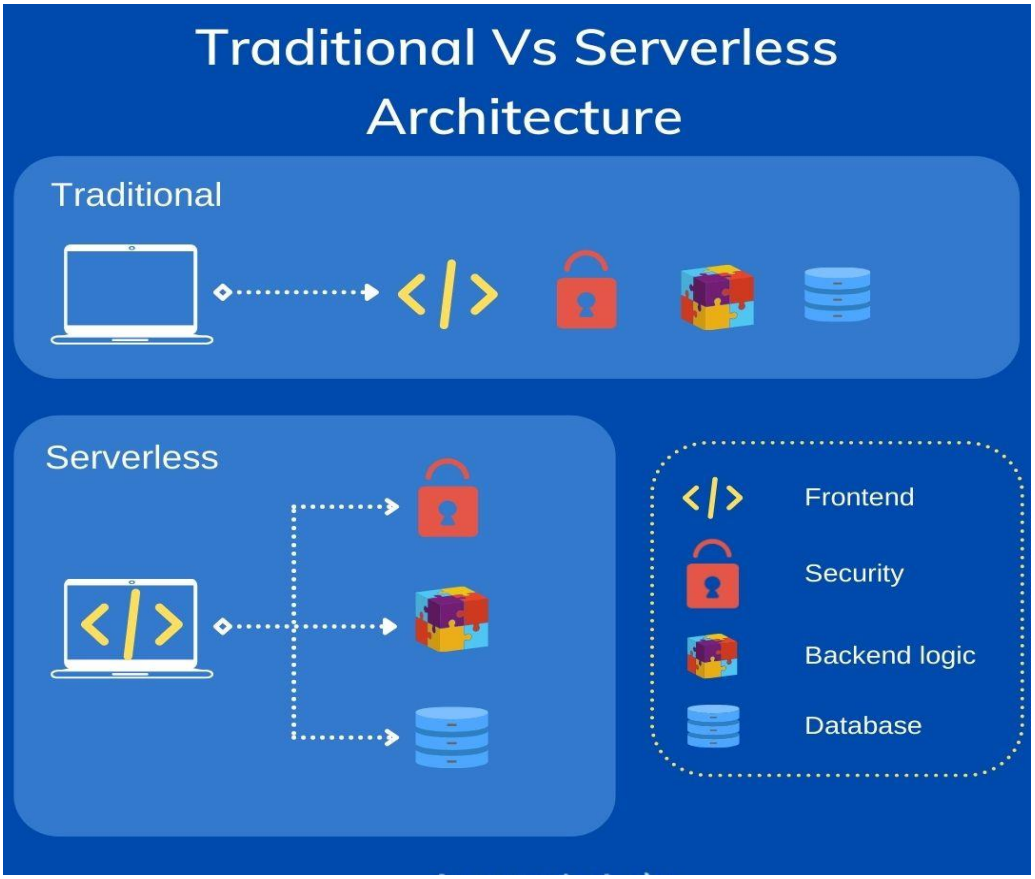
management also helps reduce operational overhead of developers concerned about patching, server maintenance and capacity planning. Serverless computing delivers many benefits, but it also comes with tradeoffs (support is limited up to an execution time, state management may be complex, and it is vendor locked in). Despite this, however, serverless computing remains an attractive choice for modern cloud native applications, allowing businesses to provision a resource with the agility, efficiency and cost effectiveness they need to operate their workloads.

Importance of the Study

This study is important because it may enable its findings to help businesses understand how serverless computing can help optimise costs. With so many companies moving to cloud technologies, organizations need to understand the financial consequences of adopting serverless architectures. Server hosted models are often inefficient which can lead to resource over provisioning due to peak loads and unnecessary spend. Moving to serverless model gives businesses the ability to pay only for the compute it uses and save on managing applications costs. We also study the effect of serverless computing for cost optimization as it lowers upfront capital expenditures and operational costs, particularly in cases with volatile or unpredictable workloads. Additionally, it discusses how serverless platforms enable scalable and flexible architecture thus bringing more efficient resource allocation. As we see cloud native applications and microservices take root and serverless computing becomes a popular technique for providing additional latency handling, it's important for organizations to understand how they can get the most from serverless computing while at the same time improve overall agility while keeping cost under control. This research will attempt to demystify many of the financial benefits of serverless computing to aid businesses in making smart decisions as they migrate to the cloud and utilize their resources. In doing so it provides a roadmap for companies to maximize their operational efficiency, minimize waste and to develop a technology infrastructure that is aligned with the financial goals of the organization.

Traditional vs. Serverless Computing Models

The traditional computing model involves reliance on physical or virtual servers, which the organizations are provisioned, managed and maintained. Specifically, businesses either own physical hardware, or lease virtual machines from cloud providers, and configure and manage them themselves in this model. Workload based provisioning is used because often the servers are over provisioned as companies must account for peak loads and have enough capacity to support traffic peaks. This can cause underutilized resources at the times when they are not needed and wasted costs to boot. However, organizations need to also handle software patches, system updates, network configurations, scaling and other day to day tasks, scaling the overhead and the operational complexity.



The serverless model however abstracts away the servers and allows developers only to deploy code as functions or microservices in the form of functions. Serverless platforms like AWS Lambda, Azure Functions, and Google Cloud Functions take away that need and hand over everything from provisioning to scaling, monitoring, and maintenance to the cloud provider. Modular code written by the developer and which is triggered on events is all that the developer needs to be concerned with. With the serverless model, companies only pay for the actual compute time that the application uses, since they're not paying for idle server capacity.

Key differences between the two models are:

1. Infrastructure Management: Traditional models require manual management of hardware or virtual machines, whereas serverless computing abstracts away all infrastructure management.
2. Scalability: In traditional computing, scaling requires manual intervention or complex configurations (e.g., load balancing, adding more servers), whereas serverless computing automatically scales resources based on demand.
3. Cost Structure: Traditional computing usually involves fixed or resource-based pricing (e.g., paying for server capacity), whereas serverless computing follows a pay-as-you-go model based on actual resource usage.
4. Operational Overhead: Serverless computing reduces operational overhead by removing the need for manual scaling, patching, and monitoring, while traditional computing often requires extensive ongoing maintenance.

Serverless computing offers greater flexibility, scalability, and cost efficiency for dynamic workloads, while traditional computing may still be more suitable for predictable and steady applications that require dedicated resources.

Literature Review

Elgamal, T., et al (2018). "Costless: Reducing cost of serverless architectures through function fusion and placement" proposes some strategies to cut the costs for serverless architectures by optimizing cost of deploying the serverless functions. Serverless computing is great for many use cases, but you will often end up with high operational costs because of many function invocations, a lot of resource utilization. The paper suggests two primary optimization techniques: On the other side, function fusion and function placement. Function fusion combines several smaller functions into a single function so as to reduce the cost of multiple executions by significantly reducing the cost of function invocations. On the other hand, function placement optimizes where functions should be deployed across the cloud infrastructure, so that functions will be placed near the data or user base, reducing latency and wasting less network utilization, ultimately saving cost. Through these techniques, organizations can optimize serverless computing costs to increase the total cost effectiveness of cloud based services without compromising scalability and performance.

Lin, C., & Khazaei, H. (2020). The key problems involved with balancing performance vs. cost in serverless computing environments are addressed in "Modeling and optimization of performance and cost of serverless applications." While serverless architectures can have many advantages such as scalability and flexibility, they are commonly accompanied by unforeseen costs related to function invocation frequency, execution duration, and resource utilization. This paper proposes models to predict cost and performance of serverless applications given workload characteristics and system configuration. We explore optimization techniques of dynamic scaling and cost aware function placement to improve cost efficiency without performance degradation. Dynamic scaling enables resources to be allocated on a real time basis as per demand to be utilised avoiding at least over provisions or underutilization. Function placement which is cost aware, is the process of placing functions in a way to reduce data transfer costs, as well as minimize function execution time. Modeling these factors with the help of appropriate optimization algorithms allows the organizations to get the best of the performance and find cloud cost optimization for their serverless application running on cloud.

Adzic, G., et al (2017). The advent of serverless computing has changed the economics and the architecture of businesses' adoption of IT infrastructure. From a financial standpoint, serverless models transform the cost structure from a capital expenditure (CapEx) upfront spent to operate to an operational expenditure (OpEx), wherein companies only pay for the usage at actual use and do not pay for the usage when the resource is not being utilised. It leads to cost savings especially for workloads with varying demand. Server less computing eliminates the need for a dedicated DevOps team as cloud provider's provisions, scales, and maintains servers, relieving businesses from core application development. Architecturally, serverless computing is a microservices approach where applications are segmented into smaller, self-containing functions that can be managed, deployed and scaled independently. It offers more flexibility and brings time to market faster. Additionally, it introduces problems in managing distributed architectures, e.g., function state management and inter service communication. Serverless computing is scalable, agile, and cost efficient; however, making the shift to such a paradigm involves rearchitecturing and adapting workflows to the paradigm — often with a dependence on cloud service providers and third party tools. In the end, serverless computing is a powerful choice for almost every business because of the economic and architectural consequences.

Shahane, V. (2022). In today's cloud environments, serverless computing provides developers with flexible and scaleable solutions, abstracting away the need for managing instances of infrastructure by writing code, instead of maintaining servers. Microservices is still a common architectural patterns that can be found in serverless applications; the applications are broken down into small, independent functions that can scale

independently. Additionally, this modular approach helps the organization become more agile, and easier to deploy. In serverless environments it is optimized by reducing cold start times, improving function execution time, and reducing resource utilization. This can be significantly improved through techniques such as using lightweight runtimes, efficient code, and caching frequently accessed data. Nevertheless, monitoring and auto scaling are crucial to be responsive to varying loads. Among the best practices for deployment in serverless computing is to use CI/CD pipelines for automated testing and deployment of your application and usage of infrastructure-as-code (IaC) tools for version control, making sure that all the proper security is in place, like access control and data encryption. If your business follows these strategies in serverless computing, performance, cost efficiency, and scalability can be reaped in cloud based applications.

Sedefoğlu, Ö., et al (2021). When deploying serverless functions, we have to consider how to optimize resource usage and pay for only what you need, not overprovision. One of the key strategies is to design the functions so they perform their function in as little execution time as possible because cloud providers tend to charge based upon execution duration and memory usage. And if functions are able to fire up quickly, this reduces costs too since cold start times, when functions are invoked after being idle, take longer. Leveraging efficient code and libraries to minimize memory and CPU consumption while optimizing the use of the resources as resource and execution happens too is another cost saving technique. Event driven triggers that are emailed on specific needs reduces useless invocations, further driving down costs. Usage patterns can be monitored and function configurations, like memory allocation, adjusted to adopt to requirements of the actual workload, helping to manage costs. Finally, there are discounted pricing options for high demand functions if you adopt reserved concurrency, so that you get dedicated resources and are spared scaling costs.

Li, Z., Tan, et al (2021). Serverless cloud computing cost optimization addresses the problem of using cloud resources efficiently to reduce expenses without compromising performance and scalability. Cloud providers charge frequently based on invocation duration and/or resource consumption, thus one important approach is to optimize for execution time and memory usage of function. Techniques such as warm-up strategies and function complexity reduction, facilitated by cloud service saturation, can help in reducing cold start latency and hence reduce costs. Efficient coding practice includes use of lightweight libraries and optimization of data processing to reduce resource usage. Service functions are triggered only when necessary, which reduces idle resources and unnecessary executions and brings down costs further. Businesses can use monitoring and adjust serverless workloads configurations according to real usage patterns, for example to adjust memory allocation or select appropriate execution timeouts. Auto scaling policies can also leverage to optimally allocate resources dynamically in conjunction with changing demand patterns, to be cost optimally operated. The strategies here can help organizations easily save a ton of money in serverless cloud environments while helping to also ensure you are getting the maximum performance and scalability.

Bardsley, D., et al (2018). Serverless performance optimization strategies are intended to reduce latency, increase execution speed and minimize resource use to enable scalability over cost. One of the key approach is the reduction of cold start times which could affect performance when the functions are invoked after it being idle. Cold start delays can be mitigated with techniques including keeping functions warm using periodic invocations, or tapping into provisioned concurrency. Function code optimization increases function efficiency by minimizing dependency, using lightweight library and algorithm performance as optimization path to reduce execution time and memory usage. Another critical strategy is fine tuning memory allocation to avoid wasting resources on too little memory or on unnecessary slow down. By

relying on caching mechanisms that cache frequently accessed data in memory or across caches distributed across a network, function speed can also be improved by eliminating repeated computations or data fetches. Using real-time cloud native monitoring tools, they are able to monitor their function performance in real-time and identify areas where they might have bottlenecks and then optimize their execution based on actual usage patterns. Collectively, these strategies improve the performance of serverless platforms while keeping cost and scaling in check.

Purpose and Scope of the Study

In this study we attempt to evaluate the role of serverless computing for cost optimization of businesses adopting cloud based solutions. As applications become more complex and larger in scale, traditional server based models tend to suffer from inefficiencies for example over provisioning of resources and high operational costs. Serverless computing, which eliminates the burden of infrastructure management, works on a pay as you go pricing model, has become a potentially optimal solution in cost optimization. The purpose of this study is to investigate how serverless architectures could decrease the amount of capital expenditures spent, use less operational overhead, and be more resource efficient by charging businesses only when the application is executed and consuming compute time. Moreover, it explores serverless computing scalability benefits, especially optimal for applications that exhibit varying or uncertain workloads to demonstrate the cost advantage of serverless computing over conventional computing models.

This study includes a detailed analysis on serverless computing platforms: AWS Lambda, Azure Functions and Google Cloud Functions and how it heats up the cost structures. Slicing around the edges: Real world scenarios tested across various industries, digging into use cases where serverless computing has greatly optimized the costs from event driven applications, API management, stateless microservices, and more. The study also discusses some of the challenges facing serverless adoption, e.g., cold start latency, state management difficulties, and vendor lock in that can compromise cost efficiency in some cases. The scope of this work is to provide a complete view of the financial and operational implications of serverless computing so that organizations have the understanding necessary to make informed decisions about serverless architecture adoption. The findings are particularly important, given that many organizations are making significant investments in cloud computing and are in the process of aligning their technology strategies with longer term financial goals.

Methodology

This study on the impact of serverless computing on cost optimization uses a mixed methods approach in which both qualitative and qualitative analysis is conducted. The first step involves conducting in depth literature review to study related work, industry reports, and case studies for serverless computing and traditional computing models. We present a theoretical foundation and context to understanding the cost optimization potential of serverless architectures. The data collection phase constitutes the second phase including the analysis of real world case studies of businesses that have already deployed serverless computing (AWS Lambda, Azure Functions, Google Cloud Functions). The cost related factors considered in these case studies include resource utilization, scalability, operation overhead and maintenance cost. Further, surveys and interviews with IT managers, cloud architects and business entities using serverless and traditional compute solutions will be conducted to obtain details on cost saving strategies, challenges and experience.

In the last phase, data analysis will be carried out by comparing run costs of workloads on serverless platforms with run costs for traditional server based models. Which also includes pricing models and operational expenditures, as well as scaling efficiency. Comparisons between both models will be made in terms of cost metrics such as Total cost of ownership (TCO), Return on investment (ROI) and resource efficiency. The results will be outlined in charts and tables to show cost variances and supply actionable actions for businesses that may aim to save on their cloud infrastructure costs.

Result and Discussion

Table 1 Impact of Serverless Computing on Cost Optimization

Factor	Serverless Computing	Traditional Computing	Impact on Cost Optimization
Pricing Model	Pay-as-you-go, based on execution time and resources consumed	Fixed costs, based on provisioned resources (server capacity)	Serverless optimizes costs by only charging for actual usage, eliminating over-provisioning.
Scalability	Auto-scaling based on demand; resources are dynamically allocated	Manual scaling, often leading to over-provisioning or underutilization	Serverless reduces costs by scaling resources automatically, optimizing usage during varying loads.
Infrastructure Management	Fully managed by the cloud provider (no need for server maintenance)	Requires manual setup, management, and maintenance of servers	Reduced operational costs with serverless, as businesses don't need to invest in infrastructure or system administration.
Operational Overhead	Minimal, with automatic updates and patches managed by the provider	High, as businesses need to handle system administration tasks	Serverless reduces operational complexity, allowing businesses to focus on code and reducing staff costs.
Resource Utilization	Efficient resource usage due to dynamic scaling	Often inefficient, with resources allocated based on peak demand regardless of actual usage	Serverless ensures resources are used optimally, minimizing waste and reducing overall costs.
Startup/Scaling Costs	No upfront costs; only pay for what is used	High upfront costs for hardware, software, and IT staff	Serverless provides a lower entry cost and flexibility, avoiding the upfront capital

			investment required in traditional setups.
Cost Predictability	Can vary based on usage patterns (e.g., traffic spikes)	Fixed monthly costs, predictable but can result in paying for idle capacity	Traditional computing offers predictability but may result in waste, while serverless offers cost efficiency during fluctuating usage but less predictability in extreme scenarios.
Use Cases	Best suited for variable workloads, event-driven applications, microservices	Ideal for stable, predictable workloads with constant demand	Serverless is more cost-effective for applications with unpredictable traffic or varying workloads, whereas traditional models are better for constant, steady demands.

On cost optimization, this compares Serverless Computing with Traditional Computing in this table. More specifically, serverless computing is described by its pricing model that—unlike traditional computing, in which fixed costs are incurred per provisioned server with no assurances on the consumption of those servers—are based on a pay-as-you-go system and charge businesses for actual resource usage (execution time and consumed resources). From scalability perspective, serverless computing is more scalable since it's auto-scaling based on demand which optimizes the use of resources during spikes in their traffic load, whereas the traditional models involve manual scaling leading to inefficient resource over-provisioning or underutilization. Regarding infrastructure management, serverless computing is entirely managed by a cloud provider as opposed to businesses having to constantly invest in or maintain physical servers for traditional systems, which comes with an ongoing cost of setup, management and maintenance. The use of serverless computing minimizes operational overhead as, for instance, the provider automatically handles updates and patches eliminating the need for IT staff; this is a cost for traditional setups. Serverless computing has an efficient utilization of resources as it allocates resources as per need but when we consider traditional computing there is a chance of inefficient utilization of resources. Serverless computing has very low startup or scaling costs with no upfront capital expenditure required as with traditional computing. With serverless models, costs can be as predictable as some serverless models or not at all due to traffic spikes, but have more predictable costs than with a traditional system with the possibility of paying for idle capacity. Serverless computing is good for variable workloads and event driven applications while traditional ones are good for constant stable workloads.

Table 2 Comparison of Cost Optimization between Serverless Computing (AWS Lambda) and Traditional Computing (Dedicated Servers)

Factor	**Serverless Computing (AWS**	**Traditional Computing**	**Impact on Cost Optimization**

	Lambda)	**(Dedicated Servers)**	
Cost Per Execution	$0.20 per 1M requests (with execution time considered)	Fixed cost per server (e.g., $100/month)	Serverless provides cost savings for small to medium applications by only charging for actual usage.
Resource Utilization	100% efficient (only resources used are billed)	Typically 20-30% underutilized during off-peak hours	Serverless optimizes resources by eliminating idle times, reducing waste.
Scaling	Auto-scaling, up to millions of concurrent requests	Manual scaling, requires IT intervention or additional servers	Serverless scales automatically based on demand, eliminating the need for manual intervention and preventing over-provisioning.
Startup Costs	None, pay as you go	High upfront costs (servers, infrastructure)	Serverless has no capital expenditure, making it more cost-effective for startups or projects with unpredictable traffic.
Operational Overhead	Minimal (no server management, auto-updates)	High (IT staff needed for server maintenance, updates)	Serverless reduces overhead by eliminating server management responsibilities, freeing resources for development.
Maintenance Costs	No ongoing server maintenance required	High, including patching, security updates, hardware replacement	Serverless eliminates maintenance costs, as the cloud provider manages infrastructure updates.
Elasticity	Elasticity built in, resources scale in real-time	Limited elasticity, requires manual intervention to scale	Serverless computing offers seamless scalability with zero manual effort, improving cost efficiency.

Cost Predictability	Usage-dependent, fluctuates with traffic spikes	Predictable but may lead to over-provisioning	While serverless may be unpredictable at times, it ensures cost efficiency by charging only for active usage, unlike fixed resource models in traditional computing.

This table provides the comparison of cost optimization between Serverless Computing (AWS Lambda) and Traditional Computing (Dedicated Servers). Serverless computing in which the charge is per actual use ($0.20 per 1 million requests) is more economical for small to medium applications than the traditional servers, in which businesses pay $100/month fixed cost regardless of use. Serverless computing is highly efficient in resource utilization because businesses only pay for the resources that are used, whereas traditional computing has upto 20 – 30% underutilization of resources during off-peak hours adding to waste of cost. In serverless computing, scaling is automatic (resources are automatically scaled to demand) while in traditional computing manual scaling is required, which can lead to scenarios like overprovisioning, or underprovisioning. In terms of costs, there is no initial investment in serverless computing, sellers charge on a pay as you go basis (as in you pay on demand) whereas traditional setups involve massive capital expenditure (expenditure for hardware and infrastructure). Serverless computing has minimal operational overheads in that the cloud provider handles server management and updates, relieving the traditional computing responsibility of an IT staff for maintenance. With serverless, there are no maintenance costs, as the provider manages infrastructure updates, traditional models cost continuously, patching, security, and hardware replacements. At last, the elasticity of serverless environments is seamless in the sense that it automatically scales according to demand and elastic scaling is not feasible in traditional systems without any interaction. With serverless computing, you can reduce costs by allowing more flexibility, usage of limited resources, and eliminating management overhead.

Conclusion

Overall, serverless computing is a revolutionary paradigm in cloud infrastructure that caters for effective cost optimization rather than the conventional computing models. Using serverless platform like AWS Lambda, Google Cloud Functions, or Azure Functions, businesses can adopt pay-as-you-go pricing model, hence pay only for the resources they are actually using, but not overprovisioning anymore. Proof of lease models empower dynamic resource allocation, enabling on demand, scalable operations where costs are relative to actual demand. By not needing to worry about such factors as servers, software updates or physical infrastructure businesses are also able to reduce personnel and maintenance costs.

In addition to that, serverless architectures optimize resource utilization through the implementation of real time scaling resources, eliminating waste and ensuring operational efficiency. Moreover, the abrogation of upfront infrastructure costs makes it easier for businesses such as startups or projects that have an unpredictable workload to get onboard. While it’s worth considering, serverless models carry with them cost unpredictability during traffic spikes or highly variable usage patterns, making traditional models still preferred for predictable, stable workloads. Likewise, ultimately serverless computing is a very cost effective solution for business with variable or event driven traffic but perhaps traditional computing still

has its place when dealing with steady consistent traffic. The results show that for many organizations, serverless computing can result in significant cost savings, operational improvements and scalability.

References

1. Elgamal, T., Sandur, A., Nahrstedt, K., & Agha, G. (2018, October). Costless: Optimizing cost of serverless computing through function fusion and placement. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)* (pp. 300-312). IEEE.
2. Lin, C., & Khazaei, H. (2020). Modeling and optimization of performance and cost of serverless applications. *IEEE Transactions on Parallel and Distributed Systems*, *32*(3), 615-632.
3. 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).
4. Shahane, V. (2022). Serverless Computing in Cloud Environments: Architectural Patterns, Performance Optimization Strategies, and Deployment Best Practices. *Journal of AI-Assisted Scientific Discovery*, *2*(1), 23-43.
5. Sedefoğlu, Ö., & Sözer, H. (2021, March). Cost minimization for deploying serverless functions. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (pp. 83-85).
6. Li, Z., Tan, Y., Li, B., Zhang, J., & Wang, X. (2021, March). A survey of cost optimization in serverless cloud computing. In *Journal of Physics: Conference Series* (Vol. 1802, No. 3, p. 032070). IOP Publishing.
7. Bardsley, D., Ryan, L., & Howard, J. (2018, September). Serverless performance and optimization strategies. In *2018 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 19-26). IEEE.
8. Sudharsanam, S. R., Namperumal, G., & Selvaraj, A. (2022). Integrating AI/ML Workloads with Serverless Cloud Computing: Optimizing Cost and Performance for Dynamic, Event-Driven Applications. *Journal of Science & Technology*, *3*(3), 286-325.
9. Kumari, A., Patra, M. K., Sahoo, B., & Behera, R. K. (2022). Resource optimization in performance modeling for serverless application. *International Journal of Information Technology*, *14*(6), 2867-2875.
10. Mahajan, K., Figueiredo, D., Misra, V., & Rubenstein, D. (2019, December). Optimal pricing for serverless computing. In *2019 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE.
11. Feng, L., Kudva, P., Da Silva, D., & Hu, J. (2018, July). Exploring serverless computing for neural network training. In *2018 IEEE 11th international conference on cloud computing (CLOUD)* (pp. 334-341). IEEE.
12. Mahmoudi, N., & Khazaei, H. (2020). Performance modeling of serverless computing platforms. *IEEE Transactions on Cloud Computing*, *10*(4), 2834-2847.
13. Eismann, S., Grohmann, J., Van Eyk, E., Herbst, N., & Kounev, S. (2020, April). Predicting the costs of serverless workflows. In *Proceedings of the ACM/SPEC international conference on performance engineering* (pp. 265-276).
14. Jarachanthan, J., Chen, L., Xu, F., & Li, B. (2022). Astrea: Auto-serverless analytics towards cost-efficiency and qos-awareness. *IEEE Transactions on Parallel and Distributed Systems*, *33*(12), 3833-3849.
15. 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.
16. Patros, P., Spillner, J., Papadopoulos, A. V., Varghese, B., Rana, O., & Dustdar, S. (2021). Toward sustainable serverless computing. *IEEE Internet Computing*, *25*(6), 42-50.

17. Cordingly, R., Shu, W., & Lloyd, W. J. (2020, August). Predicting performance and cost of serverless computing functions with SAAF. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* (pp. 640-649). IEEE.
18. Shafiei, H., Khonsari, A., & Mousavi, P. (2022). Serverless computing: a survey of opportunities, challenges, and applications. *ACM Computing Surveys*, *54*(11s), 1-32.
19. Li, Y., Lin, Y., Wang, Y., Ye, K., & Xu, C. (2022). Serverless computing: state-of-the-art, challenges and opportunities. *IEEE Transactions on Services Computing*, *16*(2), 1522-1539.
20. Munasinghe, A. K. D. M. (2021). *Optimizing costs of Serverless: Function profiling approach to optimize memory and running time of Serverless functions* (Doctoral dissertation).