IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

# Optimizing Big Data Insights with Serverless Architecture

**Manikandan M[1] and Haripriya V[2]**
PG Student, Department of MSc CS-IT[1]
Assistant Professor, School of CS & IT[2]
Jain (Deemed-to-be University), Bangalore, India
[1]manikandanm271101@gmail.com and [2]v.haripriya@jainuniversity.ac.in

**Abstract:** *Big data is the huge amount of data, which can be structured, semi-structured, or unstructured, that is required for current commercial processes. Big Data efforts and technologies are used to analyze large amounts of data in order to gain insights critical for strategic decision-making. Data size is constantly rising, reaching petabytes, exabytes, zettabytes, and even yottabytes, offering substantial management and processing issues. In practice, managing massive amounts of data involves several obstacles, such as server management, storage, clustering, and algorithm deployment. Manual intervention hampers the creation of successful Cloud-based data analysis platforms. Serverless computing provides a solution by offering clients pay-per-use backend services, reducing the need for users to manage server operations. This article describes a serverless architecture for large data analytics, including implementation, maintenance, and governance on Amazon Web Services (AWS). Furthermore, it investigates the differences between traditional and big data analytics in a serverless system.*

**Keywords:** Serverless Cloud Computing; AWS Serverless Service; AWS Lambda; Big Data Analysis; Amazon Web Service

## I. INTRODUCTION

Serverless computing is a deployment approach in which the cloud provider is responsible for running code by dynamically allocating resources. This code runs in stateless containers and can be triggered by a variety of events, including database events, HTTP requests, monitoring alerts, scheduled events, and file uploads. Essentially, serverless allows users to create and run code without having to manage servers. In a serverless environment, consumers pay just for continuous execution or performance, not for specific server components. This means that organizations who use backend services from serverless providers are billed based on their compute demand rather than having to maintain and pay for a predetermined amount of bandwidth or servers. Serverless' primary goal is to shift the attention away from infrastructure maintenance and toward application development. Previously, establishing a web application required real hardware to run a server, which was costly and time-consuming. The cloud enabled remote rental of fixed numbers of servers and server space. However, firms frequently overpaid for server capacity in order to avoid violating monthly limits, resulting in unused server space. To solve this, cloud providers offered auto-scaling solutions, but these were sometimes prohibitively expensive. Serverless computing, also known as Function as a Service (FaaS), allows developers to pay for backend services based on execution, which means they only pay for what they utilize. Serverless' primary goal is to shift the attention away from infrastructure maintenance and toward application development. Previously, establishing a web application required real hardware to run a server, which was costly and time-consuming. The cloud enabled remote rental of fixed numbers of servers and server space. However, firms frequently overpaid for server capacity in order to avoid violating monthly limits, resulting in unused server space. To solve this, cloud providers offered auto-scaling solutions, but these were sometimes prohibitively expensive. Serverless computing, also known as Function as a Service (FaaS), allows developers to pay for backend services based on execution, which means they only pay for what they utilize.


DOI: 10.48175/IJARSCT-15934

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

## II. LITERATURE SURVEY

Developers and architects specializing in serverless architecture have conducted experiments to assess several aspects including Big Data Analytics, Lambda performance, latency, scalability, source efficiency across various platform types, architecture phase, flint, and design patterns. Making Use of PyWren The IBM-PyWren reimplementation of the PyWren API, developed by Josep Sampe [1], mostly carries out MapReduce tasks. In his writing, Dimitar Kumanov [2] discusses the price, terms, and circumstances of serverless computing using AWS Lambda as well as its amazing performance. The scalability, functionality, and efficacy of the AWS, GCP, and Azure platforms were investigated by Liang Wang [3]. Sanghyun Hong [5] proposed a six-pattern layout system for creating security solutions. They present a program that embodies benefits and offers pattern assistance for each design blueprint. Kim Youngbin [6] leverages serverless data analytics with AWS and Flint Lambda. Every serverless provider tool, including Auth0 Webtask, Google Cloud Functions, Microsoft Azure Functions, IBM Bluemix OpenWhisk, Galatic Fog Gestal Laser, and AWS Lambda, has been reviewed by Theo Lynn [7]. Speaking on the shortcomings and difficulties of serverless systems, Joseph M. Hellerstein [8] concentrated on AWS Lambda. Following some function execution, Garrett McGrath [10] displayed the metrics to assess the effectiveness of various serverless computing platforms, including AWS Lambda, Azure functions, Google Cloud functions, and IBM's deployment of Apache OpenWhisk. We addressed storage, analysis, normalization, and other factors when designing a comprehensive serverless architecture for big data analytics in our study.

## III. OBJECTIVES

- To clarify the role that serverless architecture plays in resolving issues related to scalability, affordability, and flexibility in big data analytics systems.
- To give a summary of the frameworks, platforms, and serverless architecture techniques that are currently in use, with an emphasis on how they are used to big data analytics.
- To investigate several serverless architecture features and components, such as pay-per-execution models, stateless containers, and event-driven computing, and evaluate their applicability for handling massive datasets.
- To evaluate the advantages and disadvantages of serverless architecture for big data analytics, taking into account development and deployment simplicity, resource utilization, and performance.
- To determine what obstacles still need to be overcome in order to deploy serverless architecture for big data analytics, including handling data storage, guaranteeing data security, and streamlining data processing procedures
- To promote the use of serverless architecture as a scalable and affordable big data analytics solution, highlighting its potential to democratize access to sophisticated analytics capabilities.
- To provide perspectives and suggestions for future lines of inquiry targeted at augmenting the dependability, expandability, and effectiveness of serverless architecture for large-scale data analytics uses.

## IV. EXISTING MODEL

Traditional systems frequently face difficulties with scalability, resource management, and cost-effectiveness in the field of big data analytics. Serverless architecture has come to light as a potential remedy for these issues, providing a scalable and adaptable method of handling massive amounts of data. Current serverless architecture systems for big data analytics use cloud services like Microsoft Azure Functions, Google Cloud Functions, and AWS Lambda to run code in response to events without requiring server setup or management. These systems make use of models of event-driven computing, in which tasks related to data processing are initiated by a variety of events, including user requests, scheduled events, and data ingestion. Organizations can optimize expenses by only paying for the computer resources needed during the execution of data analytics operations by implementing a pay-per-execution approach with serverless architecture. Furthermore, analytics applications may easily adapt to variations in data volume and processing demands thanks to serverless systems' built-in scalability and fault tolerance. Notwithstanding these advantages, there are still issues with managing data storage, integrating serverless apps with current infrastructure, and tracking and

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

troubleshooting them. But as serverless technologies continue to develop and expand, the systems that are now in place continue to grow, providing more reliable and effective big data analytics solutions.

## V. DISADVANTAGES OF EXISTING MODEL

While there are many benefits to serverless architecture for big data analytics, there are also certain drawbacks that must be considered. The possibility of controlling data storage becoming more complex is one major disadvantage. Organizations have more control over data storage options in traditional systems, which enables them to tailor performance and cost to meet their unique needs. However, because there is no direct access to the underlying infrastructure in a serverless environment, controlling data storage becomes more difficult. Organizations may encounter constraints concerning setup and data storage choices, which could result in inefficiencies or less-than-ideal performance. Compatibility and interoperability issues may also arise when combining serverless data storage solutions with already-existing on-premises or cloud-based data repositories. Moreover, it may be challenging to forecast and manage data storage expenses due to the dynamic nature of serverless architectures, where resources are provisioned and scaled automatically in response to demand. To ensure efficiency and cost-effectiveness in serverless big data analytics deployments, organizations need to carefully consider these drawbacks and put strategies in place to mitigate them. Some of these strategies include adopting comprehensive data management policies, utilizing hybrid cloud solutions, and closely monitoring and optimizing data storage usage.

## VI. PROPOSED MODEL

Utilizing the advantages of serverless computing, the proposed big data analytics serverless architecture system seeks to effectively analyze massive amounts of data while maintaining scalability, flexibility, and affordability. The use of event-driven data processing, in which tasks related to data processing are initiated by a variety of events, including user interactions, scheduled triggers, or data import, is the fundamental component of the system. Without the need to provision or manage servers, data processing operations can be carried out in a distributed and scalable manner by utilizing serverless computing platforms like AWS Lambda, Google Cloud Functions, or Azure Functions. The system stores massive amounts of structured, semi-structured, and unstructured data using scalable data storage options including managed database services or cloud-based object storage. In order to handle complicated data processing pipelines and coordinate the execution of various serverless functions, data processing workflows are developed utilizing serverless orchestration services. With support for serverless batch processing services for batch processing jobs and serverless stream processing frameworks for real-time analytics, this allows for both batch and real-time data processing. Strong logging and monitoring systems are put in place to measure in real-time the efficiency, dependability, and expense of serverless operations and data processing workflows. Security best practices are used to guarantee the confidentiality, integrity, and compliance of data processed within the serverless architecture. These practices include encryption, role-based access control, and data anonymization techniques.Furthermore, in order to decrease costs based on actual computing resource consumption, serverless pricing models are leveraged to detect and eliminate inefficiencies in data processing workflows through the implementation of cost optimization measures. Serverless integration platforms and standardized protocols enable organizations to gain valuable insights from their data while embracing a continuous improvement culture through performance analysis, feedback gathering, and iterative serverless architecture refinement to meet changing business requirements and big data analytics technological advancements. This facilitates seamless integration with external data sources, services, and applications. By utilizing the potential of serverless computing, the suggested system for serverless architecture in big data analytics aims to transform data processing. It functions using an event-driven approach, in which different events—like data input, user interactions, or planned triggers—cause data processing activities to be initiated. Without the need to manage infrastructure, the system can dynamically allocate resources and carry out operations in a distributed and scalable manner by utilizing serverless computing services like AWS Lambda, Google Cloud Functions, or Azure Functions. Utilizing scalable data storage options, like managed database services or cloud-based object storage, is essential to the system since it makes it possible to store vast amounts of organized, semi-structured, and unstructured data effectively. Serverless orchestration services are used to orchestrate data processing workflows, making it easier to coordinate various serverless operations and handle intricate data processing pipelines.This architecture can handle a variety of

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

analytics requirements by supporting batch and real-time processing. Robust monitoring and logging methods are included into the system to assure performance and dependability. These mechanisms offer real-time insights into the cost, performance, and dependability of serverless activities and workflows. Access control and encryption are two security mechanisms that protect the confidentiality and integrity of data. In addition, cost-optimization techniques are applied to maximize resource utilization and reduce costs. Serverless integration platforms and defined protocols enable interoperability and flexibility by facilitating seamless interaction with other data sources and services. This enables businesses to gather input and iteratively develop the serverless architecture, fostering a culture of continuous improvement and enabling them to extract meaningful insights from their data. In the end, the suggested approach gives businesses the ability to effectively, economically, and scalable realize the full promise of big data analytics.

## VII. ADVANTAGES OF PROPOSED MODEL

Significant benefits are offered by serverless architecture for big data analytics applications. First off, it provides unmatched scalability, distributing resources dynamically in response to demand and guaranteeing that analytical procedures can manage massive data volumes without experiencing performance deterioration. This elasticity is especially useful in situations when data volume varies randomly. Furthermore, serverless architecture encourages cost-effectiveness by removing the requirement for installing and maintaining fixed infrastructure and only billing businesses for the compute resources utilized to process data. Comparing this pay-per-execution strategy to conventional server-based methods can save a significant amount of money. Moreover, serverless architecture improves flexibility by relieving developers of the responsibility of maintaining infrastructure and allowing them to concentrate entirely on writing code. This shortens time-to-market by quickening development cycles and facilitating the quick deployment of analytics apps.Furthermore, because serverless platforms manage resource provisioning, scaling, and fault recovery automatically, they offer fault tolerance and high availability by default. This guarantees that analytics operations run continuously and don't require human involvement. All things considered, serverless architecture simplifies big data analytics workflows and provides scalability, cost-effectiveness, flexibility, and dependability, which makes it a desirable choice for businesses looking to effectively extract insights from massive datasets.

Significant benefits are provided by serverless architecture for big data analytics applications. First off, by intelligently allocating resources based on demand, it offers unmatched scalability, guaranteeing that analytics processes can effectively handle massive volumes of data without experiencing any performance loss. This flexibility helps organizations adjust resources up or down as needed, especially in situations where data volume fluctuates randomly.

Additionally, serverless architecture encourages cost-effectiveness by implementing a pay-per-execution paradigm in which businesses are only billed for the computational resources used to process data. Comparing this to conventional server-based methods, there is no longer any requirement for installing and maintaining fixed infrastructure, which results in significant cost reductions. Furthermore, serverless architecture increases flexibility by absolving developers of the need for infrastructure management. Developers can shorten time-to-market by speeding up development cycles and quickly deploying analytics applications when they just concentrate on creating code. Moreover, serverless platforms automatically manage resource provisioning, scaling, and fault recovery, hence providing fault tolerance and high availability by default. This increases reliability by guaranteeing that analytics operations run continuously without the need for human intervention. All things considered, serverless architecture offers scalability, cost-effectiveness, flexibility, and dependability while streamlining big data analytics workflows. These advantages make it a desirable choice for businesses looking to facilitate agile development and deployment techniques and effectively extract insights from massive datasets.

## VIII. METHODS

Amazon Web Services (AWS) created Amazon S3, or Amazon Simple Storage Service, a cloud storage solution that is well-known for its quickness, ease of use, and scalability. S3's simple interface makes it possible for users to store and retrieve any amount of data with ease, from any location on the internet. It provides developers with access to a fast, dependable, scalable, and affordable storage infrastructure that powers e-commerce websites for Amazon. Because of its many capabilities, S3 is a top option for businesses looking for scalable and effective data storage solutions for a range of workloads and applications.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

AWS Glue is an Extract, Transform, and Load (ETL) service that simplifies the loading and processing of data for analytical purposes. It is provided by Amazon Web Services (AWS). The AWS Management Console's AWS Glue feature makes it incredibly easy for users to develop and carry out ETL activities, greatly streamlining the data integration process. Within its Glue Data Catalog (GDC), the service automatically profiles data. GDC is a metadata storage repository for all data assets. The GDC is a thorough resource for managing metadata since it contains crucial information like table definitions and data location.

Amazon Web Services (AWS) offers a solution called AWS Athena that lets data analysts do interactive queries on data that is stored in Amazon S3 in the public cloud environment. Interestingly, Athena runs as a serverless query service, so analysts don't have to worry about maintaining any underlying infrastructure. AWS Athena does not require data to be put into the service or modified beforehand, in contrast to standard data analysis procedures. This simplified method greatly streamlines the analytics process for customers by making it faster and simpler to execute normal SQL queries to evaluate data stored in Amazon S3.

With the help of the Amazon Web Services (AWS) infrastructure, programmers and data scientists can quickly create, train, and implement machine learning models for analytical or predictive uses with Amazon SageMaker. Users can easily access data sources for analysis and exploration with SageMaker'sJupyter notebook environment. Furthermore, the service offers machine learning algorithms that are tuned and able to process massive amounts of data in a distributed setting with efficiency. This feature-rich feature set makes the machine learning workflow more efficient, enabling users to create and implement complex models quickly and easily.

Amazon Web Services (AWS) offers a solution called AWS Step Functions (SFs) that lets users plan, organize, and carry out distributed application-based workflows. Users can efficiently arrange and arrange the different parts of their apps in a sequential fashion by using Step Functions. The service makes it simple to visualize and control the workflow by providing a graphical terminal that shows the application components as sequential phases. Step Functions also make ensuring the program runs in the right order by automatically starting, monitoring, and retrying each step in the event of an issue. One important benefit of SFs is that they can be used to build programs without requiring any code changes. This means that users can rearrange or switch out components as needed, even when requirements change.

Big data processing and ingestion in real-time settings are made easier with Amazon Web Services' (AWS) completely managed offering, Amazon Kinesis. Kinesis is engineered to manage enormous volumes of streaming data from various sources, including operations logs and financial transactions. It has the capacity to analyze hundreds of terabytes of data each hour. With the use of this capacity, businesses can effectively collect, handle, and evaluate streaming data as it is produced, providing quick insights and useful intelligence. Businesses can create scalable, responsive apps that use real-time data to inform decisions and improve operational effectiveness with Amazon Kinesis.

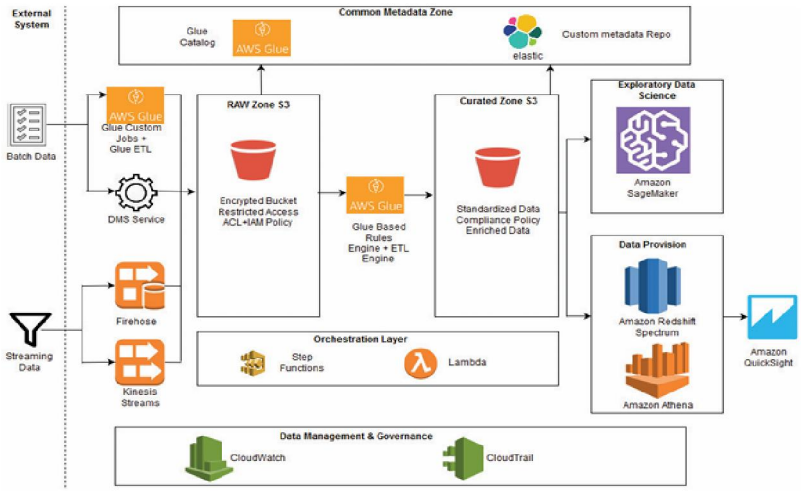


Figure1

As we move to the second tier, the Curated zone, data standardization becomes more important. It is essential to have this layer in place to ensure uniformity and consistency because data can come in different formats and from different

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

sources. Standardizing date-time formats from several time zones is one example. Before moving on to the next phase, data curation is essential to guarantee data accessibility and integrity.

In the third layer, the Exploratory zone, the aim is to uncover hidden patterns and insights within the data. This phase involves conducting exploratory analysis to identify correlations, trends, or anomalies that may not be immediately apparent. Much like exploring uncharted territories, this zone is where experimentation and discovery take place to reveal valuable insights.

The operationalization of the data for use in business applications is the last focus of the fourth layer, which is referred to as the Provisioning zone. Data is prepared and optimized here to power different organizational applications and processes. By ensuring that data is easily accessible and usable, this provisioning enables organizations to get valuable insights and facilitate decision-making.

These four levels, which provide the required framework for data input, standardization, exploration, and operationalization, essentially constitute the core of a strong data lake architecture. Similar to how every room in a house has a certain function, every layer in the data lake framework adds to the data ecosystem's overall effectiveness and usefulness.

## XI. CONCLUSION

Finally, it can be said that the serverless platform is very important for data computing in the twenty-first century. Processing big data quickly and affordably is becoming increasingly important as its volume grows dramatically. Our architectural framework demonstrates the entire procedure of gathering and archiving data from various sources, then preparing the data with AWS Glue to facilitate processing. Additionally, we present the processing and analysis stages using serverless components, which provide a less difficult substitute for complex traditional systems. Big data analytics can be carried out with the least amount of expense and greatest effectiveness by adopting serverless architecture, opening the door for future navigation over the enormous ocean of data. It is clear that serverless platforms provide various benefits that are essential for the future of big data analytics in addition to the architectural framework that has been provided. First of all, because of their scalability, increasing data volumes may be handled without the need for infrastructure provisioning or manual intervention. Because of its scalability, businesses can effectively adjust to changing data demands and always operate at peak efficiency.

Additionally, serverless architectures encourage economy of scale by implementing a pay-per-use paradigm in which resources are used only when the application is running. Because there is no longer a need for an initial infrastructure investment, companies of all sizes may now afford to utilize advanced analytics capabilities without having to pay excessive prices.

Furthermore, analytics solutions can be developed and implemented more quickly because to the flexibility and agility provided by serverless systems.

Without having to worry about managing infrastructure, developers can concentrate on creating analytical models and producing code, which speeds up time to market and promotes innovation.

Furthermore, the dependability of big data analytics procedures is guaranteed by the serverless architectures' intrinsic fault tolerance and high availability. Organizations can rely on their analytics processes to maintain smooth operations even in the face of unforeseen obstacles, thanks to automatic retries and integrated error management systems.

All things considered, the use of serverless architecture in big data analytics is poised to completely transform how businesses extract knowledge from data. Serverless platforms, with their scalability, affordability, agility, and dependability, open the door to a world in which everyone can make decisions based on data, fostering innovation and competitive advantage in the digital era.

## REFERENCES

[1] Y. Kim and J. Lin, “Serverless Data Analytics with Flint,” IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2018–July, pp. 451–455, 2018.

[2] L. Wang, M. Li, Y. Zhang, T. Ristenpart, and M. Swift, “Peeking Behind the Curtains of Serverless Platforms,” 2018 USENIX Annu. Tech. Conf. (USENIX ATC 18), pp. 133–146, 2018.

[3] G. Adzic and R. Chatley, “Serverless computing: economic and architectural impact,” pp. 884–889, 2017.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 4, March 2024

[4] G. McGrath and P. R. Brenner, "Serverless Computing: Design, Implementation, and Performance," Proc. - IEEE 37th Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2017, pp. 405–410, 2017.
[5] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, "A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a Service) Platforms," Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom, vol. 2017–Decem, pp. 162–169, 2017.
[6] J. Sampé, G. Vernik, M. Sánchez-Artigas, and P. García-López, "Serverless Data Analytics in the IBM Cloud," pp. 1–8, 2018.
[7] J. M. Hellerstein et al., "Serverless Computing: One Step Forward, Two Steps Back," vol. 3.
[8] E. van Eyk, A. Iosup, S. Seif, and M. Thömmes, "The SPEC cloud sgroup's research vision on FaaS and serverless architectures," no. Section 2, pp. 1–4, 2017.
[9] D. Kumanov, L.-H. Hung, W. Lloyd, and K. Y. Yeung, "Serverless computing provides on-demand high performance computing for biomedical research," 2018.
[10] S. Hong, A. Srivastava, W. Shambrook, and T. Dumitraș, "Go Serverless: Securing Cloud via Serverless Design Patterns," USENIX Work. Hot Top. Cloud Comput., 2018.
[11] B. Wagner and A. Sood, "Economics of Resilient Cloud Services," Proc. - 2016 IEEE Int. Conf. Softw. Qual. Reliab. Secur. QRS-C 2016, pp. 368–374, 2016.
[12] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-dusseau, and R. H. Arpaci-dusseau, "Serverless Computation with OpenLambda 1 Introduction 2 Lambda Background 3 Lambda Workloads," USENIX Work. Hot Top. Cloud Comput., 2016.
[13] M. Villamizar et al., "Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures," Serv. Oriented Comput. Appl., vol. 11, no. 2, pp. 233247, 2017.
[14] D. A. Kumari and N. Tejaswani, "Vector Quantization for Privacy during Big Data Analysis and for Compression of Big Data," vol. 4, no. 4, pp. 421–428, 2015.
[15] "Why Serverless?" [Online]. Available: https://serverless.com/learn/overview/. [Accessed: 30-Mar-2019].
[16] O. Alqaryouti and N. Siyam, "Serverless Computing and Scheduling Tasks on Cloud : A Review," pp. 235–247.
[17] "What is Serverless Architecture? What are its Pros and Cons?" [Online]. Available: https://hackernoon.com/what-is-serverless architecture-what-are-its-pros-and-cons-cc4b804022e9. [Accessed: 30-Mar-2019].
[18] "Serverless Computing – Amazon Web Services." [Online]. Available: https://aws.amazon.com/serverless/. [Accessed: 30-Mar 2019].
[19] "What is Serverless Architecture and why should you care ?"
[Online]. Available: https://medium.com/@anandujjwal/what-is
serverless-architecture-and-why-should-you-care-bcf83069eb38. [Accessed: 30-Mar-2019].
[20] M. Chan, "Serverless Architectures: Everything You Need to Know," Thorn Technol., Mar. 2017.
[21] R. Miller, "AWS Lambda Makes Serverless Applications A Reality," TechCrunch, Nov. 2015.
[22] "What Is Serverless Computing? | Serverless Definition | Cloudflare."
[Online]. Available: https://www.cloudflare.com/learning/serverless/what-is-serverless/. [Accessed: 30-Mar-2019].
[23] "Cloud Object Storage | Store & Retrieve Data Anywhere |
Amazon Simple Storage Service." [Online]. Available: https://aws.amazon.com/s3/. [Accessed: 30-Mar-2019].
[24] "What Is AWS Glue? - AWS Glue." [Online]. Available: https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html.