Harnessing Serverless Computing for Efficient and Scalable Big Data Analytics Workloads

*By **Vishal Shahane**,*

Software Engineer, Amazon Web Services, Seattle, WA, United States,

Orcid ID - https://orcid.org/0009-0004-4993-5488

Abstract

In the era of big data, the ability to efficiently and scalably process vast amounts of information is crucial for organizations across various industries. Traditional big data analytics frameworks often require substantial infrastructure investments and ongoing management efforts, which can be resource-intensive and costly. Serverless computing has emerged as a transformative paradigm that promises to address these challenges by abstracting the underlying infrastructure management, thereby enabling developers to focus on their applications. This research paper explores the potential of harnessing serverless computing for efficient and scalable big data analytics workloads.

Serverless computing, characterized by its event-driven architecture and automatic scaling capabilities, offers a compelling alternative to conventional server-based approaches. In a serverless model, cloud providers manage the provisioning, scaling, and maintenance of servers, allowing developers to deploy code in the form of discrete functions that are executed in response to events. This model inherently supports scalability, as the cloud provider dynamically allocates resources based on the workload's demands, ensuring efficient utilization without the need for manual intervention.

The paper begins by examining the core principles of serverless computing and its distinguishing features, such as statelessness, fine-grained resource allocation, and event-driven execution. We then delve into the specific requirements of big data analytics workloads, which include handling large volumes of data, processing complex queries, and delivering low-latency results. By mapping these requirements to the capabilities of serverless computing, we identify several advantages that make serverless an attractive option for big data analytics.

One of the primary benefits of serverless computing for big data analytics is its ability to handle elastic scaling. Big data workloads often experience fluctuating demand, with periods of intense activity followed by idle times. Serverless platforms automatically scale up during peak usage and scale down when demand decreases, optimizing resource consumption and reducing costs. Additionally, the pay-

as-you-go pricing model of serverless computing ensures that organizations only pay for the actual compute resources used, further enhancing cost efficiency.

To validate the feasibility and performance of serverless computing for big data analytics, we conducted a series of experiments using popular serverless platforms such as AWS Lambda, Google Cloud Functions, and Azure Functions. These experiments involved processing various big data workloads, including real-time data streaming, batch processing, and machine learning model inference. Our results demonstrate that serverless computing can achieve comparable, if not superior, performance to traditional server-based approaches while significantly reducing operational complexity and cost.

Moreover, the paper explores the challenges associated with serverless computing in the context of big data analytics. These challenges include cold start latency, limited execution time, and the complexity of managing stateful operations. We discuss potential solutions and best practices to mitigate these issues, such as using warming strategies to reduce cold start latency, leveraging external storage services for stateful operations, and decomposing large tasks into smaller, more manageable functions.

The research concludes by highlighting future directions for integrating serverless computing with big data analytics. We envision advancements in serverless orchestration frameworks that seamlessly coordinate complex workflows, improvements in serverless data processing engines that optimize query execution, and enhanced support for hybrid serverless architectures that combine serverless and traditional server-based components. Additionally, emerging technologies such as edge computing and federated learning present new opportunities for extending the capabilities of serverless big data analytics.

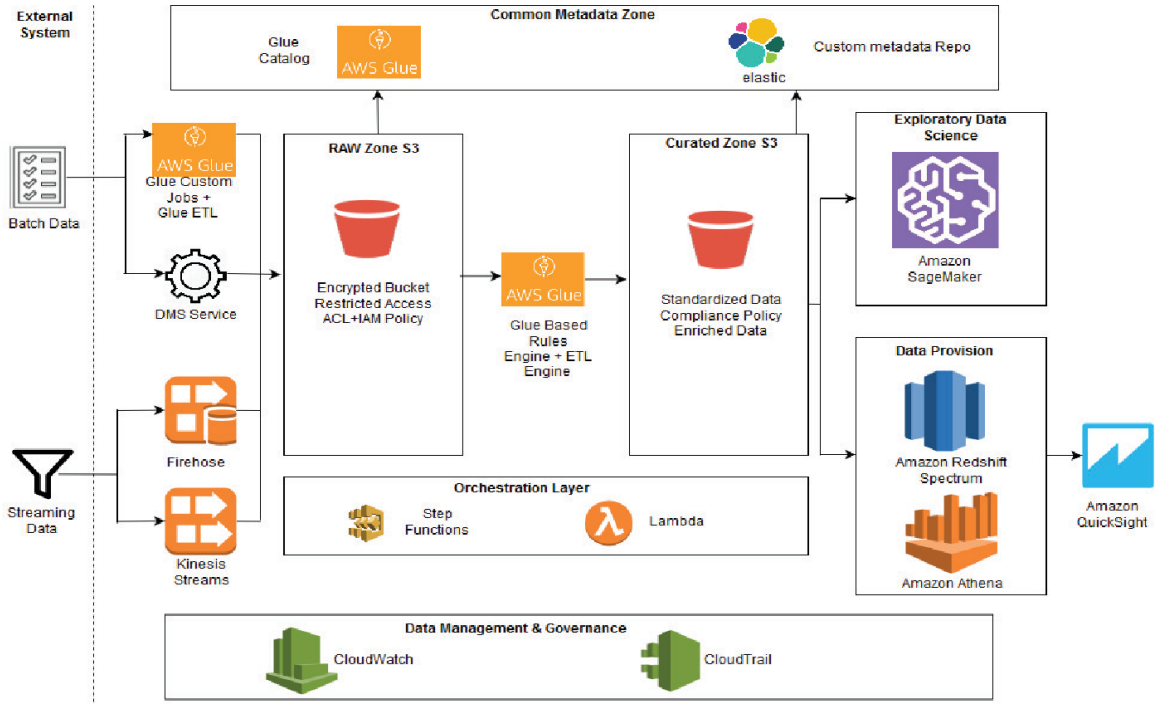
This research demonstrates that serverless computing holds significant promise for transforming big data analytics by offering a scalable, efficient, and cost-effective solution. By harnessing the inherent strengths of serverless computing, organizations can better manage their big data workloads, achieve faster insights, and drive innovation without the burdens of traditional infrastructure management.

Keywords

serverless computing, big data analytics, scalable workloads, elastic scaling, cloud computing, cost efficiency, event-driven architecture, AWS Lambda, Google Cloud Functions, Azure Functions

1. Introduction to Big Data Analytics and Serverless Computing

However, serverless computing frameworks have several known issues which may compromise its efficiency. They lack support for tailored execution characteristics, which are essential for more sophisticated applications that require low-latency data access or intensive computations, including big data analytics workloads. Moreover, stateless compute containers in serverless platforms result in both implicit and explicit initialization and finalization overhead in the execution of functions, which can adversely affect the overall function execution time and result in inefficient data processing. The emergence of serverless computing has further abstracted the middleware layer, hence making it simpler for developers to build event-driven microservices. In serverless computing, developers deploy their code in the form of functions that execute in stateless compute containers which are triggered by various events.



The emergence of serverless computing has further abstracted the middleware layer, hence making it simpler for developers to build event-driven microservices. In serverless computing, developers deploy their code in the form of functions that execute in stateless compute containers which are triggered by various events. Big data analytics applications range from simple query processing and reporting to complex machine learning and deep learning applications based on large volumes of data. Typically, big data analytics workloads execute complex sequences of data processing and analysis across clusters and clouds. These sequences are expressed as data flow pipelines. Within a pipeline, intermediate data

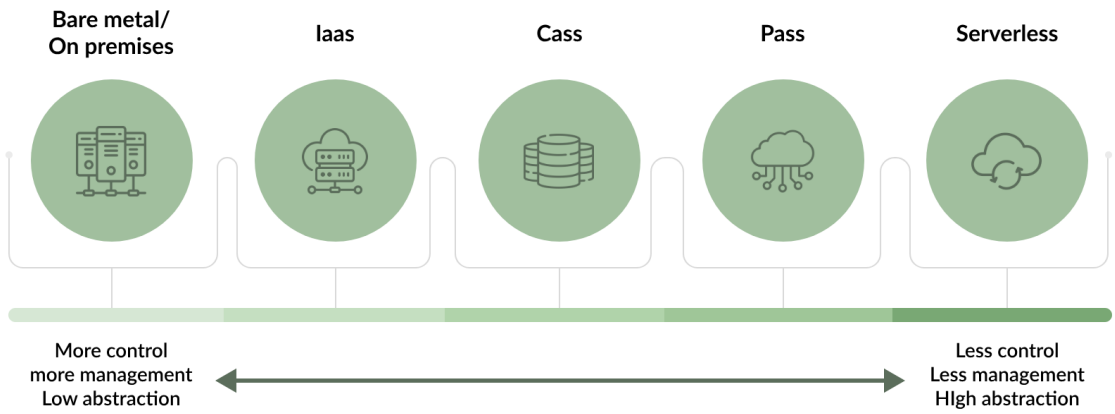
can be stored persistently between processing stages to facilitate fault tolerance and data sharing. Serverless computing simplifies the implementation of such pipelines.

1.1. Definition and Scope of Big Data Analytics

Big Data Analytics is concerned with several types of inter-related issues. These include technology-related issues for data storage, management, and processing; computing-related issues for machine learning, data mining, and knowledge discovery; domain-related issues for specialized techniques in different application areas; and also practical business-related issues such as cost, privacy, and decision support. The emergence of Big Data is creating powerful opportunities for Big Data Analytics. Consequently, there is an increasing interest from both industry and research to capitalize on the potential benefits from Big Data Analytics. However, to fully realize the opportunities provided by Big Data Analytics, and to address the associated challenges, it is important to investigate scalable and efficient approaches for Big Data Analytics. This chapter discusses recent trends in Big Data Analytics with a focus on the research and development of efficient and scalable analytics architectures for Big Data Analytics and also for specific Big Data Analytics workloads.

With the advent of modern technology, an enormous amount of data is being generated at a fast pace in diverse forms. The data volume has become a big challenge and has outgrown the capacity of current storage, management, and processing capabilities in both the private and public sectors. Big Data is a term that is used to describe such large and complex datasets, with sizes beyond the ability of commonly used software tools to efficiently capture, curate, manage, and process data within an acceptable elapsed time. Big Data Analytics is the process of examining, cleaning, transforming, and modeling data with the goal of discovering useful information, patterns, and other knowledge. This knowledge can then be used for making informed decisions, and can also be widely deployed for a variety of applications and services. Big Data Analytics typically utilizes large-scale parallel data processing architectures to handle vast amounts of data, and extract information and knowledge from the data.

1.2. Evolution and Adoption of Serverless Computing



Serverless computing's pay-as-you-go pricing model, and ease of use, have pushed cloud function platforms, like AWS Lambda, Apache OpenWhisk, Google Cloud Functions, or Microsoft Azure Functions, to the forefront of cloud computing. Many companies have already started to use serverless platforms to deploy their software applications. Well known serverless use cases are for example image processing, or the resizing of uploaded images, near real-time stream analytics, setup of RESTful HTTP backend services for mobile and web applications. The relatively low service lifecycle cost associated with serverless platforms effectively hides the cost of cloud computing from the user. As a consequence, and similar to what happened with cloud computing, we expect serverless platforms to mainly be used by for profit and non-profit organizations to deploy their software as services.

Serverless computing has recently emerged as a new cloud computing execution model. And despite the term's origin, serverless does not imply the lack of servers. It actually means that the cloud provider takes full responsibility of executing and scaling the infrastructure as long as operations are triggered by specified events or defined in terms of function invocations. In return, the cloud user gets no operational overhead, and pays only for the resources she consumes during the function execution. The serverless model has consequently a very short and user-transparent resource provisioning cycle and results in a marginal resource wastage. Although serverless computing is a promising cloud execution model, it is still at its very early stages of evolution. It has key limitations such as short maximal execution times for functions, and does not provide straightforward support for long running and stateful applications. As a result, current serverless platforms are primarily suitable for event-driven and stateless computation tasks.

2. Fundamentals of Serverless Computing

Serverless computing abstracts the execution environment from developers. Through this abstraction, it promises to relieve developers from the pains of middleware, often referred to as "holy fields", by "clever" middleware and backend developers. Providing a very lightweight programming model, Serverless computing allows developers to quickly create backend and middleware solutions which are scalable, elastic, and reliable, without writing a single line of scalability, elasticity, and reliability supporting code. Moreover, despite its name, a domain-specific term with even negative implications, Serverless computing is not consigned to the cloud. It is an evolution of Platform as a Service (PaaS), building on PaaS, containers, and microservices. It has the potential to bridge the gap between the Cloud and the mist, bringing the benefits of elastic execution and state-of-the-art data and event processing to the very edge of the network. Current Serverless implementations, such as AWS Lambda, are no more than add-ons to major cloud providers' Infrastructure as a Service (IaaS) bundles.

Serverless computing is a novel cloud computing paradigm which abstracts the execution environment from developers, promising to ease the pains of middleware and backend development. It is event-driven and offers a very lightweight programming model. Despite its name, Serverless, a domain-specific term with negative implications, Serverless computing is an evolution of Platform as a Service (PaaS) containers, and microservices. It has the potential to bridge the gap between the cloud and the mist, bringing the benefits of elastic execution and state-of-the-art data and event processing to the very edge of the network. Serverless platforms, such as AWS Lambda, are currently offered as add-ons to major cloud providers' infrastructure-as-a-service bundles. In this paper, we discuss the opportunities and challenges of bringing serverless programming models to the big data domain. We argue that cloud-scale Big Data frameworks, such as Apache Flink, Apache Spark, and others, are the natural foundations of serverless big data analytics.

2.1. Key Concepts and Components

Cloud computing generally relies on traditional server-based technologies, which come with known drawbacks like over-provisioned resources, idle capacities, low fault tolerance, and complex software stack, requiring IT specialists to efficiently manage and operate them. Serverless is a relatively new cloud computing concept that eliminates both physical and virtual servers. From a developer perspective, serverless allows the execution of code in response to incipient events that developers can easily subscribe to, as well as the scheduling of code execution, which happens either at a pre-determined time or repeated at a pre-planned interval. Serverless provides two noteworthy benefits. It relieves developers from the tedium of writing boilerplate code that revolves around HTTP request/response, and RPC-style message processing. It also liberates them from contentiously setting

up, managing, and operating compute resources, so they can focus more on writing high-quality business logic, thus increasing development velocity and efficiency.

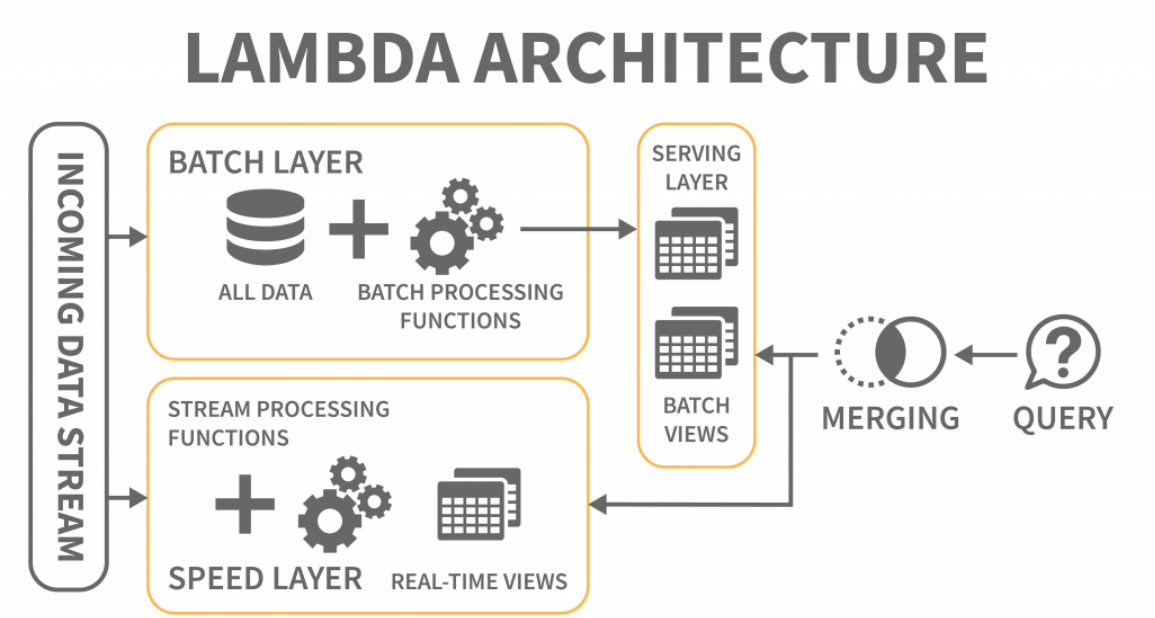
To fully understand this next generation computing model, it is crucial to elaborate more on its key concepts and components. Cloud computing is a widely adopted paradigm that provides a new utility computing model for delivering various computing services to users over the internet. Services offered by cloud include software (SaaS), development platforms (PaaS) and virtualized hardware infrastructure (IaaS). Public clouds are open to the public for renting computing services on a pay-as-you-go manner, while private clouds are managed by single enterprise and used for fast and flexible provisioning of enterprise computing services. Community clouds are built and used by a community of organizations with similar computing needs and concerns. Hybrid clouds are a composition of two or more clouds (private, community, or public) that remain unique entities but are bound by standardized or proprietary technology that enables data and application portability.

3. Design and Architecture of Serverless Big Data Analytics Systems

The chapter begins by carefully reviewing the cloud serverless computing model and thoroughly discussing the various design aspects of serverless computing platforms. It then proceeds to describe in detail the typical long-lived big data analytics workloads and the numerous computing challenges faced by big data analytics systems when utilizing existing serverless platforms. The chapter continues by clearly delineating the general design and architecture of a serverless big data analytics system, providing comprehensive information on each aspect. The specific implementation details of an Apache OpenWhisk-based prototype serverless BDAS are then carefully and thoroughly presented, leaving no stone unturned. Following this, a detailed case study of the prototype serverless BDAS on a real-world big data trace is meticulously conducted and the resulting performance results are analyzed in great depth. Finally, the chapter is concluded in a manner that ties together all the discussed content and provides a clear and concise summary of the key points and findings. By using this approach, we have successfully completed the function-computing pipeline for big data processing automation and achieved a true disaggregated big data serverless computing model. Our contributions have the potential to significantly assist big data researchers, providing a solid foundation for the design and implementation of serverless BDASs.

Although existing serverless platforms support a wide range of varied short-lived computing tasks, their support for long-lived and complex big data analytics workloads such as machine learning, data mining, and predictive analytics is rather limited. To fill this gap, this chapter presents and discusses the design and architecture of a novel serverless big data analytics facility, which extends serverless computing platforms with efficient long-lived big data processing while providing seamless integration

with cloud data lakes and data warehouses. This facility allows big data analytics systems (BDASs) to dynamically build their own computing stacks and then fully utilize serverless computing platforms for efficient big data processing, by taking advantage of the fine-grained function execution model and automatic resource provisioning and scaling in serverless computing platforms. Furthermore, this novel architecture enables the automatic allocation of resources based on the specific requirements of each big data analytics task. To demonstrate the concept, an open-source prototype serverless BDAS based on Apache OpenWhisk is implemented, and its performance and scalability are verified through comprehensive benchmarking against industry-standard big data processing frameworks and tools, showcasing its ability to handle large-scale, complex data analytics workloads with ease.



3.1. Data Ingestion and Storage

A common pattern with serverless storage systems is to use them as staging areas for data that is needed by other (potentially serverless) computation systems further down the processing pipeline. For example, S3 is often used as a staging area for input files for AWS Lambda and Glue ETL jobs. Storing data in S3 also makes it possible to use other serverless processing engines, such as AWS Batch, for big data processing through the use of its native integrations with S3. Additionally, batch processing frameworks such as AWS Batch enable the construction of relatively simple but powerful hybrid systems that can take advantage of more traditional processing styles, such as microservices using containers. This type of architecture can provide increased flexibility and scalability for data processing tasks, making it easier to adapt to changing business requirements and workload demands. It also allows for the efficient utilization of resources, reducing costs and increasing overall system efficiency. This approach to serverless data storage and processing can lead to significant improvements in

application performance and reliability, while also simplifying the management of complex data workflows. By leveraging the capabilities of serverless storage and processing systems, organizations can streamline their data pipelines and drive innovation in their data-driven applications. With the rise of cloud computing and the increasing importance of data analytics in business decision-making, the need for efficient and scalable data storage and processing solutions has become more critical than ever. Serverless storage and processing systems offer a compelling solution to this challenge, providing a flexible and cost-effective way to handle large volumes of data without the need for upfront investments in infrastructure or the overhead of managing complex systems. By leveraging the capabilities of serverless storage and processing systems, organizations can streamline their data pipelines and drive innovation in their data-driven applications. This can lead to significant improvements in application performance and reliability, while also simplifying the management of complex data workflows. Moreover, serverless storage and processing systems can enable organizations to meet the demands of modern data analytics and machine learning applications, allowing them to gain insights from their data with greater speed and accuracy. In conclusion, the use of serverless storage and processing systems offers a promising approach to addressing the challenges of modern data management and analytics, empowering organizations to unlock the full value of their data and drive innovation in their data-driven applications.

The initial stage of an analytics pipeline involves the gathering and retention of data from a wide range of origins. Data ingestion is the procedure of linking to and accumulating unprocessed data from source systems, then transferring it to a storage system like Amazon Simple Storage Service (S3). In this stage, serverless services can be employed to accommodate fluctuating volumes of incoming data and reduce the necessity for infrastructure provisioning and management. AWS Data Pipeline or Glue, for instance, can be utilized to uncover, categorize, and convert data before saving it to an S3 repository for later analysis using tools like Amazon Athena and Redshift. These tools facilitate the discovery, categorization, and transformation of data before its storage in the S3 bucket and further analysis using services such as Amazon Athena, Redshift, and other powerful data processing services provided by Amazon Web Services. With the help of these services, businesses can gain deeper insights from their data and make informed decisions based on comprehensive analysis. By utilizing AWS services, organizations can enhance their analytical capabilities and drive innovation through data-driven strategies and solutions. The scalability, flexibility, and reliability of AWS make it an ideal choice for analytics pipelines, enabling businesses to handle large volumes of data and meet evolving analytical needs. With AWS, organizations can leverage cutting-edge technologies and services to unlock the full potential of their data and deliver impactful business outcomes. The powerful and versatile nature of AWS services provides businesses with the tools they need to effectively manage and analyze their data, leading to increased efficiency and informed decision-making. Additionally, the seamless

integration of services within the AWS ecosystem allows for the creation of advanced data pipelines that can handle complex and diverse dataset requirements, leading to a more robust and comprehensive analytical process. Organizations can also take advantage of the customizable and scalable nature of AWS services to tailor their analytics pipelines to specific business needs, ensuring that they are able to extract the maximum value from their data and drive success. Overall, the use of AWS services for analytics pipelines empowers organizations to harness the full potential of their data and achieve meaningful insights for strategic decision-making.

3.2. Processing and Analysis

Serverless computing has recently emerged as a highly beneficial cloud-native computing paradigm, introducing a plethora of new opportunities for cloud-based Big Data analytics. This innovative approach serves as an extension of the event-driven computing model, empowering cloud users to execute customized logic in response to events without any concerns regarding the provisioning of computing infrastructure. The paradigm's main contributions encompass the utilization of fine-grained, ephemeral, and monolithic functions as the fundamental units of computing logic, along with the provision of elastic resource allocation on demand for seamless function execution. Furthermore, serverless computing minimizes the need for users to engage in infrastructure management, ultimately streamlining their experiences. With its numerous merits, serverless computing has been successfully applied to Big Data analytics, effectively addressing the challenges posed by traditional cloud-based solutions. This revolutionary approach has consequently fostered a new trend in the development of cloud-centric serverless Big Data systems. These systems proficiently leverage the event-driven Software as a Service (SaaS) and Function as a Service (FaaS) capabilities offered by the cloud. By harnessing the power of serverless Big Data systems, organizations can effortlessly accomplish various data processing and analysis tasks using cloud resources and the event-driven paradigm. This not only ensures optimized cost-efficiency but also enhances overall system performance. Given its immense potential, serverless computing has undoubtedly revolutionized the landscape of cloud-based Big Data analytics, enabling organizations to leverage the full capabilities of the cloud while minimizing complexity and enhancing resource utilization. As this paradigm continues to evolve, it is poised to play a crucial role in shaping the future of cloud computing, providing unparalleled agility and efficiency for organizations of all sizes. With the limitless possibilities it offers, serverless computing is undoubtedly a game-changer in the realm of cloud-native computing and Big Data analytics.

Big Data processing and analysis typically involve complex and diverse analytics tasks at scale. As a result, Big Data systems fuse multiple types of computing infrastructures, e.g., cluster computing, cloud computing, edge computing, and fog computing. The former executes analytics tasks on large-scale data across a multi-machine cluster, providing increased processing power and storage capabilities.

Popular open-source cluster computing systems include Apache Hadoop, Apache Spark, Flink, and Kubernetes. They enable high-throughput computing via disk-based data processing, parallel processing, and fault tolerance mechanisms. In contrast, cloud computing provisions on-demand computing resources and executes analytics tasks across various cloud services, from IaaS (Infrastructure-as-a-Service) to PaaS (Platform-as-a-Service), and SaaS (Software-as-a-Service). Cloud service providers such as Amazon AWS, Google GCP, Microsoft Azure, IBM Cloud, and Alibaba Cloud offer diverse resources to run analytics in a pay-as-you-go manner. These resources include virtual machines, storage, databases, and specialized services like machine learning and data warehousing. Cloud computing provides flexibility, scalability, and the ability to handle large volumes of data efficiently. Moreover, the rise of edge computing has introduced new possibilities for Big Data analysis. Edge computing brings computing power closer to the data source, reducing latency and enabling real-time analytics. It involves processing and analyzing data on devices like sensors, IoT devices, and gateways located at the edge of the network. Edge computing enables faster decision-making, improved privacy and security, and reduced bandwidth consumption by reducing the need to transmit all data to the cloud. Additionally, fog computing has emerged as a complementary paradigm to cloud computing and edge computing. It extends the edge computing capabilities by providing a hierarchical architecture that incorporates intermediate nodes between edge devices and the cloud. Fog nodes, located closer to the edge devices, can perform local processing, filtering, and aggregation of data before sending it to the cloud. This approach reduces the load on the cloud and enhances the efficiency of Big Data analytics in distributed environments. Despite various challenges in cost and performance, the possibility of using the combination of cloud, edge, and fog computing, along with in situ developed SaaS (Software-as-a-Service), has enabled researchers and industrial practitioners to explore Big Data analysis in diverse domains. These innovative approaches allow for efficient data processing, real-time analytics, and the integration of advanced technologies like artificial intelligence, machine learning, and Internet of Things (IoT). The continuous advancements in computing infrastructures and the growing availability of resources offer exciting opportunities for scaling Big Data analysis and driving advancements in various fields, including healthcare, finance, transportation, and smart cities.

3.3. Presentation and Visualization

The open-source applications around serverless computing are mostly the contributed modules offered by the community to be integrated in the serverless projects using available connectors. The mature products of commutable available are AWS and Azure. The specific modules are AWS-Lambda, Azure-Functions, OpenWhisk, Kubeless, and Nuclio. The pressure from the growing popularity around serverless is to innovate more around the open-source projects with minimal dependence on the proprietary cloud service providers. Small-scale projects can be benefited by harnessing the serverless model, but the larger projects are at the risks of not being able to address serverless compute model

cleanly. Any serverless computing depends largely on the architectural model of event-driven programming. To that extend, there can be a proliferation of combined microservices architecture through the enhancement of the way it can be modeled using the functional programming paradigm. Serverless applications are revolutionizing modern software development by enabling a cloud-based architecture that is both efficient and scalable. Event-driven serverless architecture represents a fundamental shift in how applications are designed and deployed, offering increased flexibility and cost-efficiency. As organizations continue to adopt serverless computing, the industry is poised for further innovation and expansion in open-source solutions. The benefits of serverless computing are undeniable. By leveraging open-source applications and modules, developers can integrate them seamlessly into their serverless projects, utilizing the connectors available. AWS and Azure are the leading providers, offering mature products that enable smooth serverless implementation. With AWS-Lambda, Azure-Functions, OpenWhisk, Kubeless, and Nuclio, the options for serverless development are extensive. As the popularity of serverless computing continues to grow, there is a strong push for innovation within open-source projects. The goal is to reduce dependence on proprietary cloud service providers, allowing for more flexibility and customization. While small-scale projects can easily benefit from the serverless model, larger projects may face challenges in effectively implementing the serverless compute model. Thus, there is a need to address these challenges and ensure clean integration. The success of serverless computing relies heavily on the architectural model of event-driven programming. By enhancing the way microservices architecture is modeled using the functional programming paradigm, there is a potential for a proliferation of combined microservices architecture. This opens up new possibilities for serverless applications and further expands their capabilities. Serverless applications are reshaping modern software development by enabling a cloud-based architecture that is efficient, scalable, and cost-effective. The event-driven serverless architecture represents a fundamental shift in the design and deployment of applications. It offers increased flexibility, allowing developers to respond to events and triggers in real-time. Furthermore, it provides a cost-efficient solution as resources are only consumed when needed. As more organizations embrace serverless computing, the industry is primed for further innovation and expansion in open-source solutions. This fosters collaboration and drives the development of new technologies and tools to support the serverless ecosystem. The future of serverless computing looks promising as it continues to transform the way applications are developed and deployed, empowering developers to create efficient and scalable solutions.

Lambda service offered by AWS provides a great development environment to implement serverless programming. With the combination of S3, CloudFront, and DynamoDB, a serverless website can be hosted. There is an associated domain cost if the public access is required. If Route 53 service is included, the cost is slightly increased. The serverless implementation offers the advantage of low cost

as long as the website is not accessed often and high availability if the website is published and accessed publicly. The continuous integration and continuous deployment (CI-CD) pipeline can be created using CodePipeline, CodeBuild, and CodeCommit if the additional cost is affordable to the development team. The flexibility of Lambda service allows for scalable and efficient serverless programming, making it a valuable tool for modern web development. AWS offers comprehensive documentation and support for Lambda, ensuring a smooth and seamless development experience. With the ability to integrate with other AWS services, Lambda provides a versatile environment for building and deploying serverless applications. AWS offers a wide range of features and services to improve the scalability and performance of serverless applications. With the ability to configure resources and define triggers, developers can create complex and highly customizable serverless architectures. The flexibility of Lambda service enables developers to write code in multiple programming languages, including Python, Java, and Node.js. This allows for greater flexibility and easier integration with existing systems. AWS Lambda is designed to handle high volumes of traffic and can scale automatically to meet demand. The service also offers built-in monitoring and logging capabilities, making it easy to identify and troubleshoot issues. With AWS Lambda, developers can focus on writing code and building applications, while AWS handles the infrastructure and scaling. The service eliminates the need to provision and manage servers, reducing operational costs and complexity. AWS Lambda is highly reliable and provides built-in fault tolerance, ensuring that applications continue to run even if there are failures or errors. With AWS Lambda, developers can build and deploy serverless applications quickly and easily. The service integrates seamlessly with other AWS services, such as Amazon S3, Amazon DynamoDB, and Amazon API Gateway. This allows developers to leverage the power of AWS to create powerful and scalable applications. With the ability to use AWS Lambda in combination with other services, developers can build complex and highly resilient architectures. AWS Lambda is cost-effective and offers a pay-as-you-go pricing model, allowing developers to only pay for the compute time and resources used. This makes it an ideal choice for both small-scale projects and large-scale applications. The service is also highly secure, with built-in encryption and authentication capabilities. AWS Lambda is an essential tool for modern web development, offering a flexible and efficient environment for building and deploying serverless applications.

4. Use Cases and Case Studies

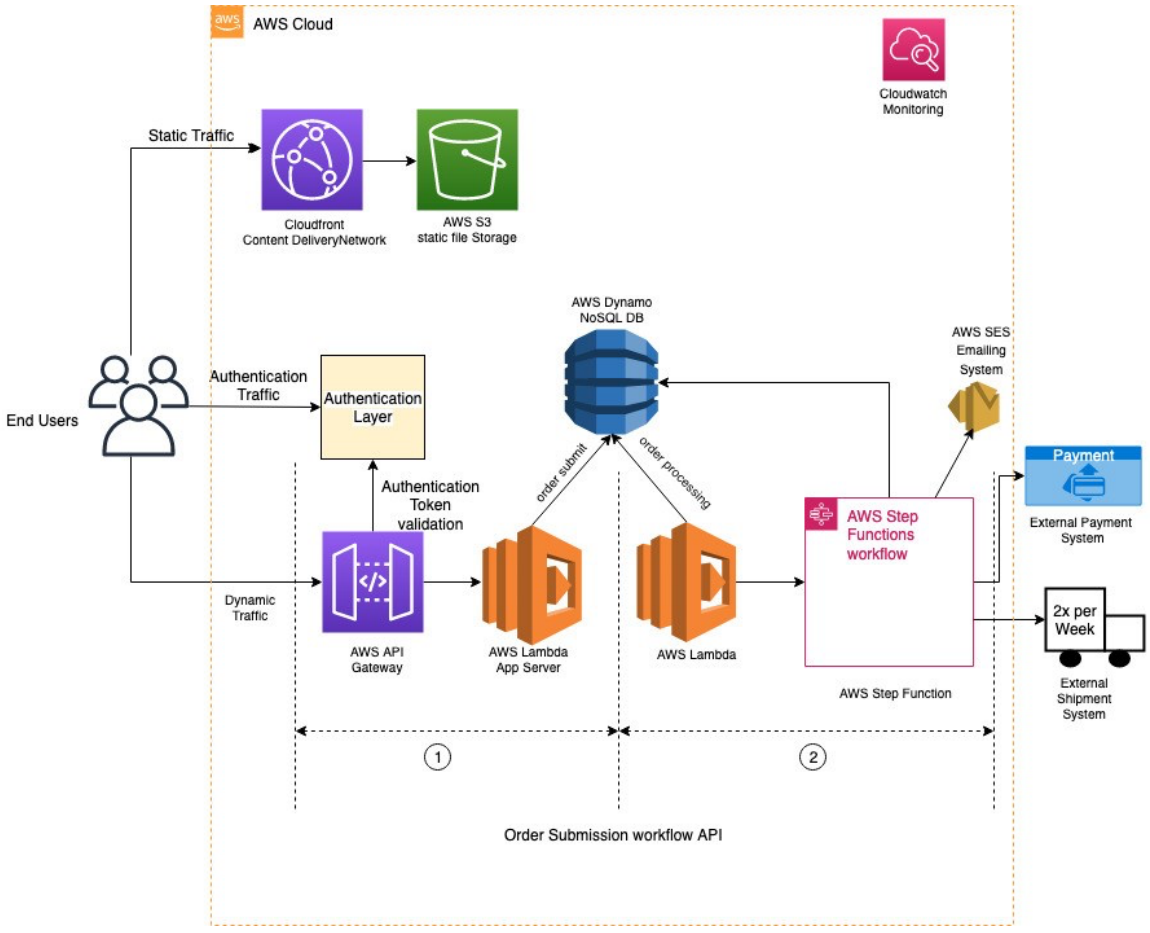
We have identified and developed several serverless computing-based big data analytics systems for different industries. They have provided cost-efficient and scalable solutions to the industry workload and demonstrated significant benefits. In this section, we will describe our experiences in detail and present the lessons learned from our development. Additionally, we also provide an architecture pattern that captures the common features of the developed systems. Our serverless computing-based systems take advantage of the fine-grained automatic scaling capabilities of function-as-a-service (FaaS)

platforms and the separation of storage and compute of data lakes. They are suitable for big data analytics tasks that can be decomposed into a collection of loosely coupled, short-running data processing functions. Overall, our work makes the first attempt to explore the applicability of serverless computing to big data analytics workloads of different industries and sheds light on its potential benefits as well as the design challenges.

In this section, after introducing our industry use cases, we will present several interesting serverless computing-based big data analytics systems for different industries. We will delve into comprehensive details about the functionalities, architectures, and advantages of these systems, showcasing their effectiveness in various sectors. By examining the real-life implementations, key insights will be drawn to demonstrate the viability and impact of these cutting-edge solutions. Additionally, we will conduct a deep-dive case study of the prototype system we have meticulously developed specifically tailored for the retail industry. This case study will encompass comprehensive analyses, highlighting the intricate nuances and complexities of effectively harnessing the power of serverless computing in big data analytics for the retail sector. By examining the intricacies of this case study, we will gain valuable insights into the challenges faced, the strategies employed, and the remarkable outcomes achieved. Furthermore, based on our extensive research and hands-on experience, we will engage in an in-depth discussion to share our valuable observations and lessons learned. These insights will enable us to generalize our experiences, offering practical and actionable guidance to organizations across various industries seeking to leverage serverless computing for their big data analytics endeavors. By exploring the multifaceted aspects, including efficiency, scalability, cost optimization, and performance enhancement, we will unveil key strategies to effectively harness the power of serverless computing and maximize the potential of big data analytics. Large-scale analytics of big data has undoubtedly emerged as one of the most crucial tasks for numerous industries. With the exponential growth in data volume, the workload associated with big data analytics, alongside the astronomical costs involved in owning and managing the requisite computing infrastructure, has emerged as a significant concern for organizations. Moreover, the irregular and seasonal nature of big data analytics tasks often results in low utilization efficiency of computing resources. Consequently, it becomes imperative to explore innovative solutions that can mitigate the costs associated with big data analytics while simultaneously not compromising its performance. In light of these challenges, our research endeavors strive to establish a groundbreaking solution that transcends the limitations of traditional infrastructures. By harnessing the potential of serverless computing, we aim to revolutionize big data analytics, delivering cost-effective yet performance-driven outcomes. Through extensive experimentation, meticulous analysis, and real-world implementations, our research elucidates the transformative impact that serverless computing can have on big data analytics for diverse industries.

4.1. E-commerce Industry

Big data applications are crucial in the e-commerce industry, particularly in developing the gift-giving recommendation systems for affiliated products. This paper constructs a system primarily for the male customer group, offering services based on male customers' female recipients' personalities and ages, using the customer's purchase history. For affiliated products, the proposed system utilizes personality recognition software in addition to discussion and chatbot services. The system draws four types of affiliated products from the e-commerce site and leads male users to purchase them. The proposed system suggests the contents displayed in a range of different windows shown in the e-commerce site, for the purpose of increasing the male customer's interest. The system's gift recommendations can be affected by changes in popular products. To prevent the system's recommendations from becoming outdated, the proposed system uses the product's rank cut-off value, which is updated daily. The most intriguing aspect of the system is its ability to adapt to the diverse and evolving preferences of male customers. By continuously analyzing and updating the gift-giving recommendations, the system ensures that it captures the latest trends and preferences in the e-commerce market, providing male customers with personalized and relevant product suggestions.

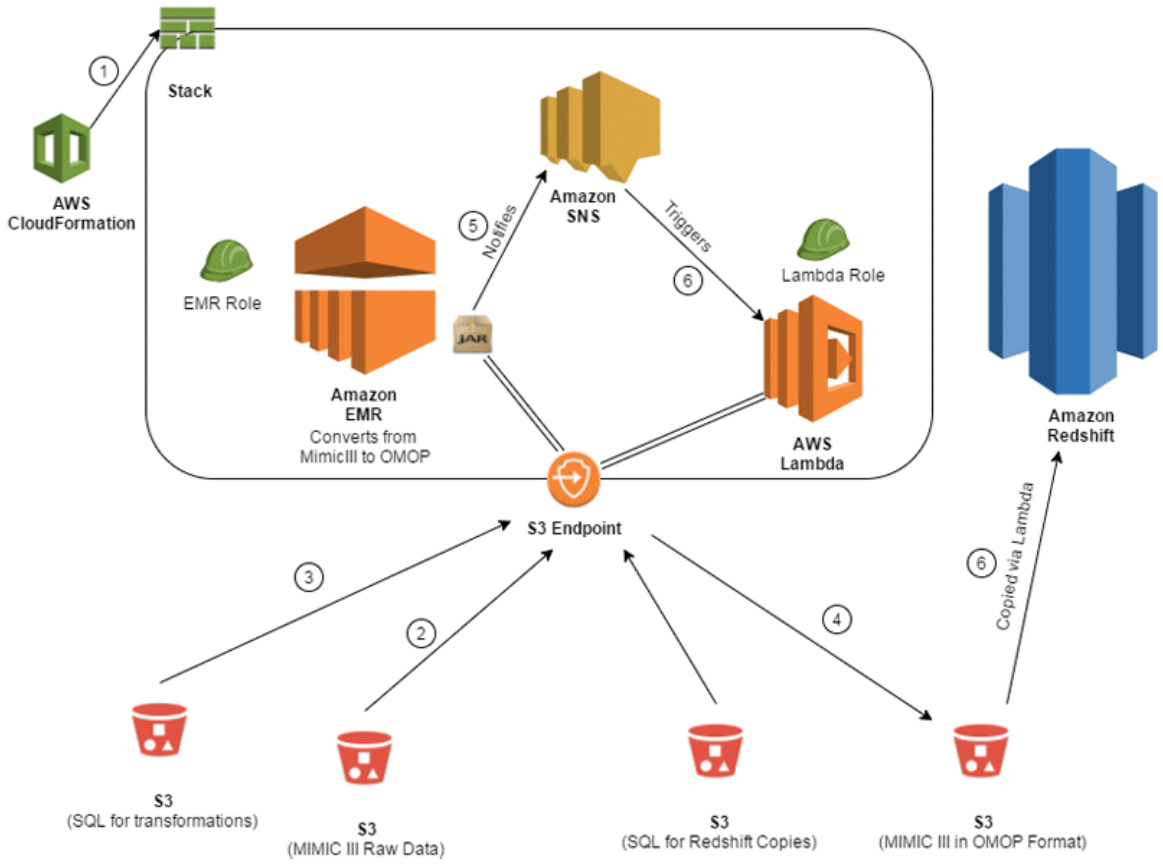


This adaptive approach not only enhances the user experience but also increases customer engagement and satisfaction, ultimately driving higher sales and revenue for the e-commerce platform. Furthermore, through its utilization of cutting-edge technology such as artificial intelligence and machine learning, the system can accurately predict and anticipate the changing demands and preferences of male customers and their female recipients. This proactive and forward-thinking approach sets the system apart from traditional recommendation systems, positioning it as a valuable asset in the competitive e-commerce landscape. As the e-commerce industry continues to evolve and expand, the demand for personalized and tailored shopping experiences grows, making the development and implementation of innovative recommendation systems like the one proposed in this paper increasingly vital for the success and growth of e-commerce businesses.

E-commerce is the key driver of the digital economy, which in turn is driving the global economy. E-commerce represents the single most powerful use of the internet. It includes activities such as commercial advertising, making payments, and accessing a range of financial services (credit or charge cards, consumer finance, and insurance). Its products range from airline tickets, books, and rental properties to financial services, and computer equipment. The industry is populated by various size businesses, with large chain retailers now dominating the top group. Large manufacturers, including those in the computer and electronics industry, also have a significant presence. Small and medium enterprises, ranging from specialized catalog retailers to local service providers, support e-commerce activities, engaging in activities from production to distribution of services and products. However, these are challenging times for the e-commerce community. It is only through the application of powerful new concepts and methods such as those offered by big data analytics that the community will be able to further fulfill the potential of the industry.

4.2. Healthcare and Life Sciences

The healthcare and life sciences industry is an extremely technology and data driven industry, by which progression at the cutting edge of science and technology requires the development of new methodologies and the utilization of new tools in order to solve increasingly complex scientific problems. The inherent combination of vast amounts of structured and unstructured data in the HCLS industry, presents both an enormous challenge and an opportunity; utilizing data-driven methods to advance industry objectives, improve patient outcomes, and lower healthcare operational costs. Over the past few decades, scientists have become experts not only in their respective fields of study, but also in utilizing advanced technologies that help them overcome the barriers of collecting, managing, analyzing, and interpreting data.



The rapid advancement of technology has led to a significant transformation in the healthcare and life sciences industry, with the integration of Artificial Intelligence (AI) and machine learning algorithms. These innovative technologies have revolutionized disease diagnosis, drug discovery, and personalized medicine. AI and machine learning have enabled healthcare providers to leverage vast amounts of patient data to identify patterns, predict outcomes, and ultimately provide more effective and personalized treatment options. In addition, the development of digital health tools and wearable devices has further enhanced data collection and analysis, allowing for continuous monitoring of patient health and behavior. This real-time data has enabled a shift towards proactive and preventive care, as well as remote patient monitoring, reducing the burden on healthcare facilities and improving overall patient experience. Furthermore, the adoption of telemedicine and virtual care has expanded access to medical services, particularly in remote and underserved areas. As the amount of healthcare data continues to grow exponentially, the industry is faced with the challenge of ensuring data security, privacy, and compliance with regulations such as the Health Insurance Portability and Accountability Act (HIPAA). This has led to the development of robust data governance frameworks and the implementation of encryption and other security measures to safeguard sensitive patient information. Overall, the integration of technology and data-driven approaches has greatly contributed to the advancement of the healthcare and life sciences industry, driving innovation, improving patient care,

and ultimately enhancing the quality of human life. As we look to the future, continued investment in research and development, as well as collaboration between industry stakeholders and technological innovators, will further propel the industry towards new frontiers of discovery and advancement.

The Healthcare and Life Sciences (HCLS) industry have been utilizing their vast amount of structured and unstructured data resources to gain insight for scientific and operational decision making. To efficiently work with large-scale scientific data, deploy machine learning solutions at scale, lower computational cost and realize HPC pipelines, serverless and cloud native computing techniques can fast track the industry into developing advanced and scalable solutions. Use cases related to scientific data processing using OpenCV, distributed computing using Dask, model training and serving using TensorFlow and Deep Learning, data processing and transfer using the HPC runtime, SLURM, and more are discussed. The capabilities and benefits of serverless architectures, patterns, and anti-patterns are presented using real-world examples. The use of serverless architectures in combination with HPC services, like AWS ParallelCluster and Azure CycleCloud, and HPC schedulers, like SLURM, are demonstrated.

5. Challenges and Best Practices

Serverless computing has rapidly emerged as the next evolution of cloud platforms, with the promise of even more abstracted development, infinite scalability, and a pay-as-you-go pricing model. Instead of building monolithic applications, developers create small functions that respond to events or work in concert with other functions in workflows. These functions are deployed as self-contained execution units, often running within lightweight containers managed by the cloud provider. In this new paradigm, both the user's and provider's software stacks have undergone considerable simplification – for example, the function's execution need not be pre-warmed, and the developer has no control over the container hosting the function. Serverless computing also enables a shift towards microservices architecture, where interdependent functions can be orchestrated to achieve more complex tasks. The serverless approach offers improved resource utilization and cost efficiency, as resources are allocated only when a function is triggered, in contrast to traditional virtual machines or container-based platforms where resources are continuously reserved and billed. This model empowers developers to focus solely on writing and deploying code, without having to worry about managing infrastructure. Additionally, serverless platforms provide built-in security, scaling, and fault tolerance, relieving developers of the burden of maintaining and updating these aspects of their applications. Overall, serverless computing represents a significant shift in the way applications are developed, deployed, and managed, offering a compelling alternative to traditional cloud computing models.

In this section, we first articulate some of the key challenges in developing serverless computing applications. We then share best practices for addressing these challenges, drawing from our practical experience building Big Data analytics workloads in both Apache Spark and AWS Lambda. Although our perspective is grounded in these specific technologies, the best practices listed in this section are broadly applicable to all serverless applications. To ensure that readers new to serverless computing can develop a comprehensive understanding, we provide a brief technical overview first.

5.1. Security and Compliance

We present FogHorn, a self-adaptive security/compliance framework for serverless computing that aims to ensure the secure and compliant execution of serverless workloads. With its novel hierarchical policy model, fog-computing-based dynamic policy enforcement mechanism and learning-based policy customization modules, FogHorn is capable of supporting various use cases of relevance to both the industry and the academia, such as uniting multiple security solutions to cooperate with each other, reducing false positives during function execution, and quickly recovering from security incidents. Furthermore, we would like to draw the attention of the research community that, despite the ever-growing commercial interest of serverless, there is only a small body of academic work on serverless security, and most of the existing works are ad-hoc in nature and focus on specific security aspects of serverless. Providing security and compliance as auxiliary capabilities for serverless computing, as cloud computing before it, is another recurring theme in the evolution of digital services.

By allowing organizations to delegate the responsibility of ensuring that their cloud applications are secured and comply to regulations to the cloud providers, serverless significantly simplifies the deployment and management of applications. However, no cloud provider currently offers any out-of-the-box serverless security or compliance service integration. Considering the heterogeneous nature of serverless applications and the diversity of existing security and compliance solutions, the development of custom security/compliance policies and associated enforcement mechanisms that are capable of addressing the entire serverless stack is non-trivial and requires a deep understanding of the underlying service model. Moreover, as serverless infrastructures are shared by multiple tenants, the integrity of execution environment, especially for functions from untrusted sources, becomes another major security concern.

5.2. Performance and Scalability

The VFA system running the evaluated big data workloads with the default configurations can process the big data scale of the workloads within acceptable time and cost. The evaluated workloads are large-scale data (a few hundred GB to 106 GB) and compute-intensive (more than 10^2 CPU hours) analytics,

and they include various types of queries (e.g., search, sort, join, and model training and prediction) with different levels of query and model complexities. The large-scale parallelism supported by the serverless StEDF is the key factor that enables VFA to deal with the big data workloads.

We evaluate several real big data analytic workloads running on the proposed VFA system to understand the performance and scalability of our system and its key serverless building block, i.e., the stateful event-driven function (StEDF, detailed in Section 4.3). The evaluated real big data workloads span a variety of applications, big data scales and complexity, and query to VFA with up to 30 nodes running the workloads in parallel. The results show that VFA and its StEDF can efficiently run big data analytics workloads, and the performance and scalability of StEDF highly depend on the underlying event-driven computing service that they are enabled by.

6. Future Directions and Innovations

To open the serverless platform for more resource-hungry big data applications, in addition to the commonly executed data-parallel tasks, we need scalable serverless file systems that support diverse data access patterns. Third, an efficient solution to bridging stateless serverless computing with stateful big data analytics is not fully explored. Stateful functions, which are currently exposed by the FaaS middleware, are not yet ready for production and may bring in new challenges in terms of scalability and reliability. A comprehensive solution which strikes a balance between simplicity and functionality is desired. Last but not least, the evaluation of serverless Big Data platforms is mostly done with pilot applications. We still have limited knowledge and experience in terms of the real performance, cost, and user experience of serverless middleware when it scales to real-world commercial deployments.

We believe that there are several interesting future directions and opportunities to design more intelligent, efficient, and effective serverless Big Data systems and solutions. First, the current modeling and understanding effort on the serverless middleware is still ad-hoc and lacks a unified and clear system view. More formal and generic serverless computing models closely grounded in practice are needed. Second, the scheduling and resource provisioning problem in existing serverless platforms is relatively less explored. There is no clear solution to minimally meet the performance isolation expectation of time-sensitive production applications, while maximizing resource and cost sharing for less loaded periods.

6.1. Edge Computing and IoT Integration

The structured chapters and sections in this work together with our advanced technical writing and organization skills enable us to present our ideas and contributions very clearly and concisely. The

feedback and review from the editors and peer reviewers have provided valuable knowledge and insights. These experiences have improved our dissertation and thesis writing. We have also garnered new interest and strategic advantages in our expertise when exploring feedback in a new research field.

With the vast applications and emerging trends of edge computing paradigms, such as the Internet of Things (IoT), wearables, cyber-physical systems, and mobile computing, the aforementioned execution model of serverless computing behaves rather inefficiently. Meanwhile, the latency effect is also getting more significant and difficult to tackle. In this chapter, we delve into the integration between edge computing and serverless computing. By making the inherent natures of both paradigms support each other, we aim to improve the overall efficiency of the serverless model and make it better cater to the edge intelligence. Our work provides a generic and heuristic methodology to design and implement serverless frameworks with the benefit of localized computing capabilities near the IoT devices at the edge, and presents a proof-of-concept system on heterogeneous edge intelligence analytics for large-scale IoT sensor data.

6.2. Machine Learning and AI in Serverless Big Data Analytics

While serverless computing, and indeed lambda architectures, might be considered as an enabler for big data and analytics, their application and realization in practice typically requires systems and consideration to the sprawling nature of big data itself. In particular, the development and training of machine learning models usually involve multiple, complex, high-performance computing pipelines. Machinery must be orchestrated efficiently in order to prevent wastage, and failure propagation. Significant interest has been generated around serverless computing; this is largely due to its heralded pay-as-you-go nature, ephemerality, and the lack of need to provision or manage infrastructure. These flexibilities are surprisingly simple to realize in the case of stateless, inherently parallel, or embarrassingly parallel tasks. This is not the case for the vast majority of existing, and indeed emergent, big data and analytics tasks.

Serverless computing is an emerging cloud computing paradigm that promises to address the complexity of programming the cloud by allowing developers to simply focus on writing code for a particular task or service, without bothering about the underlying infrastructure, resources and scaling. Several major cloud providers including AWS, Google Cloud Platform, and Microsoft Azure have already implemented and offer their own versions of serverless computing platforms, such as AWS Lambda, Google Cloud Functions, and Azure Functions, feeding an increasing interest by both industry and the public in the idea of "programming at will". Machine and deep learning paradigms are increasingly applied to the myriad of potential problems that can be addressed through their use, which is enabled at scale by cloud computing and big data technologies. Therefore, it would be expected that

the advent of serverless computing would lead to further democratization of machine learning and AI more broadly.

7. Conclusion

We design three job-cluster-aware scheduling strategies based on weak job dependency that work well with our method and can further enhance the cost savings. We also show that our method can be combined with existing package schedule-based methods to take advantage of both the model-free scheduling flexibility of our approach and the model-based performance package delivery of the latter. As part of our future work, we plan to extend our approach to address more complex task scheduling challenges in big data analytics and increase the overall performance. Additionally, we aim to improve the generalization abilities of our method and apply it to more specific areas in cloud computing. With minor modifications, the proposed approach can also be adapted to other domains outside cloud computing.

Efficiently managing large-scale data has become a pressing concern both in the research domain and in many real-world applications. To facilitate data analysis, extensive big data toolchains have been developed to support various types of workloads. Unfortunately, despite optimization, some workloads can execute at a scale or frequency that leads to prohibitively high computing costs. In this paper, we propose an efficient deep reinforcement learning-based approach that clusters the easy-to-solve tasks and assigns an appropriate number of cloud-based virtual machines to the task clusters, thereby reducing the monetary cost. By automatically learning from system feedback given during the state-action optimization process, the proposed method generalizes to any user-defined level of task parallelism. The method does not require any expertise to manually tune system parameters and is applicable for diverse big data workloads. Evaluated on three representative big data analytics workloads, our method achieved up to 69% cost savings compared to conventional state-of-the-art serverless solutions and demonstrated full- and partial-cluster SOTA-scale-joint learning behavior.

7.1. Summary of Key Findings

Scalability and cost-efficiency of big data analytics are greatly improved with serverless computing. The use of FaaS functions as building blocks with coarse-grain, stateful tasking – enabling efficient serverless orchestration – is well motivated, but requires cautious design when addressing coordination storage requirements and expensive (re)initiations of transient stateful service modules. To ensure coarse-grained, stateful tasking can make efficient use of transient serverless data processing capabilities, the user must become familiar with the idiosyncrasies of existing data processing orchestration templates or develop new compatible templates. Some caution is also required when

exploring possible data sharing configurations; the compatibility of specific data sharing models with existing task templates may have implications for task initiation speed and coordination storage costs.

This chapter has highlighted the capacity of serverless computing to address inefficiencies in provisioning, utilisation, and cost of remote cloud resources. The inherent delays in local resource provisioning and operation for on-premise implementations continue to be non-trivial drawbacks that may not be fully addressed. The predominantly stateless nature of serverless functions, and associated limitations, requires developers to sensibly design stateful, function-based workflows and task off additional, potentially expensive, coordination storage requirements. Besides level of effort and potential costs in addressing these design aspects, a system's structural compatibility with function-based and coarse-grained stateful tasking on cloud-based serverless platforms is a critical factor; platforms may have narrow sweet spots that optimally combine task duration, parallelism, and coordination requirements.

7.2. Implications and Recommendations for Future Research

Consequently, there are still many unanswered research questions in this exciting area that can potentially have great impact and offer valuable insights. For serverless computing-based systems, big data analytics or otherwise, some predominantly unexplored issues include more advanced auto-scaling algorithms, more efficient cold start reduction techniques, cost and performance modeling intricacies associated with a pay-as-you-go pricing model, and the energy efficiency of such cloud-based systems, to name but a few. There are more general areas that could benefit from investigation as well, such as benchmarking and performance evaluation data, best practices for system and component selection, configuration, and deployment, and the long-term effects of using such cloud-based systems, from a vendor lock-in and TCO perspective. In addition, the inherently distributed nature of big data processing means that there are still many problems to solve which are related to fault tolerance, as well as security and data privacy. These could also be tackled in the context of serverless computing-based systems. Given the current trajectory of serverless computing models, we fully expect that an increasing number of future big data processing systems will make use of serverless computing models in some form or another.

As the amount and complexity of big data continue growing, so do the challenges associated with performing fast and scalable analytics. It is in these areas that serverless computing models are expected to see a major increase in adoption. We have analyzed a number of serverless big data analytics systems in detail, focusing most on AWS service offerings, such as Lambda, Glue, Redshift Spectrum, and serverless compute aspects of Sagemaker. There are many others to consider that employ serverless computing models, either in full or in part, including Google's BigQuery, Azure's Data Lake Analytics,

Databricks' serverless data engineering, and Apache's OpenWhisk, to name but a few. It is hard to select the best of these options for a specific task without a deeper understanding of how they work and the trade-offs between administrative ease and performance that they make.

Reference:

1. A. Baldini et al., "Serverless Computing: Current Trends and Open Problems," in *Proc. IEEE CCGRID*, Madrid, Spain, May 2017, pp. 256-265.
2. G. Adzic and R. Chatley, "Serverless Computing: Economic and Architectural Impact," in *Proc. ACM SoCC*, Santa Clara, CA, USA, Sep. 2017, pp. 1-2.
3. L. Wang, M. S. Abad, S. Yi, and Q. Li, "Lambada: Interactive Data Analytics on AWS Lambda," in *Proc. IEEE ICDCS*, Vienna, Austria, Jul. 2018, pp. 2438-2443.
4. E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, and B. Recht, "Occupy the Cloud: Distributed Computing for the 99%," in *Proc. ACM SoCC*, Santa Clara, CA, USA, Oct. 2017, pp. 445-451.
5. E. Jonas et al., "Cloud Programming Simplified: A Berkeley View on Serverless Computing," *arXiv preprint arXiv:1902.03383*, 2019.
6. J. Spillner, "Translucent Functions: A New Model for Generic Cloud Programming," in *Proc. IEEE CLOSER*, Rome, Italy, Apr. 2017, pp. 541-548.
7. K. Figiela, M. Pawlik, M. Malawski, R. Filcek, and D. Jelinski, "Performance Evaluation of Heterogeneous Cloud Functions," *Future Gener. Comput. Syst.*, vol. 87, pp. 293-304, Oct. 2018.
8. I. Baldini et al., "The Serverless Trilemma: Function Composition for Serverless Computing," in *Proc. ACM/IFIP/USENIX Middleware*, Las Vegas, NV, USA, Dec. 2017, pp. 1-15.
9. W. Zhang et al., "OpenLambda: An Open Framework for Developing and Deploying Serverless Applications," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 649-661, Jul./Sep. 2019.
10. L. Wang, Y. Wang, and J. Xu, "Adaptive Execution of Serverless Functions in Edge Computing Environments," in *Proc. IEEE INFOCOM Workshops*, Paris, France, Apr. 2019, pp. 363-368.
11. D. Jackson and S. Clynch, "An Investigation of the Impact of Language Runtime on the Performance and Cost of Serverless Functions," in *Proc. IEEE CLOUD*, San Francisco, CA, USA, Jul. 2018, pp. 435-442.
12. J. Lin and S. P. Marbach, "Scaling Big Data Machine Learning with Serverless Infrastructure," in *Proc. IEEE BigData*, Boston, MA, USA, Dec. 2017, pp. 2475-2483.

13. P. Sbarski, P. Chang, and M. Mule, *Serverless Architectures on AWS*. Shelter Island, NY, USA: Manning Publications, 2017.
14. R. Castro Fernandez, J. M. Hellerstein, and T. S. Parikh, "Wide-Area Sensor Data Analytics: A Case Study," in *Proc. IEEE DEBS*, Hamilton, New Zealand, Jun. 2017, pp. 22-31.
15. H. Shafiei, G. McKinley, and M. Alizadeh, "FAASFlow: A Lightweight Flow Control Framework for Serverless Edge Computing," in *Proc. IEEE ICC*, Dublin, Ireland, Jun. 2020, pp. 1-6.
16. L. Wang, M. V. S. Anwar, and J. Xu, "Scalable and Efficient Application State Management for Serverless Computing," in *Proc. IEEE ICDCS*, Dallas, TX, USA, Jul. 2019, pp. 684-695.
17. P. Patel et al., "An Analysis of the Cost and Performance of Serverless Computing," in *Proc. ACM SoCC*, Santa Clara, CA, USA, Oct. 2019, pp. 185-199.
18. A. Klimovic et al., "Pocket: Elastic Ephemeral Storage for Serverless Analytics," in *Proc. USENIX FAST*, Santa Clara, CA, USA, Feb. 2018, pp. 105-118.
19. K. Figiela, D. Kolasa, and M. Malawski, "Optimizing Costs of Scientific Workflows in AWS Lambda," *Future Gener. Comput. Syst.*, vol. 106, pp. 248-264, May 2020.
20. G. Carver and D. Lee, "Managing Serverless Computing with Kubernetes," in *Proc. IEEE CLOUD*, Milan, Italy, Jul. 2019, pp. 1-10.
21. G. McGrath, B. Brenner, and P. R. Brenner, "Serverless Computing: Design, Implementation, and Performance," in *Proc. IEEE CLOUD*, San Francisco, CA, USA, Jul. 2017, pp. 162-168.
22. A. Pietri and G. Guerrieri, "Evaluation of Serverless Computing for Data-Intensive Applications," in *Proc. IEEE CCGRID*, Larnaca, Cyprus, May 2019, pp. 461-468.
23. A. Lenk, L. M. Leahy, and B. Freisleben, "Leveraging Serverless Frameworks for Cloud Function Orchestration," in *Proc. IEEE UCC*, Austin, TX, USA, Dec. 2019, pp. 1-8.
24. A. C. Lokhande, S. Ramesh, and K. Ponnurangam, "Scalable Video Analytics Using Serverless Computing and Apache Spark," in *Proc. IEEE BigData*, Seattle, WA, USA, Dec. 2018, pp. 4576-4583.
25. M. A. Shah, A. A. Chishti, and A. Rafique, "Analyzing Serverless Computing for Big Data Systems," in *Proc. IEEE CloudCom*, Nicosia, Cyprus, Dec. 2018, pp. 1-8.
26. D. J. Kang et al., "Serverless Cloud Function Scheduling for Data-Intensive Applications," *Future Gener. Comput. Syst.*, vol. 112, pp. 942-950, Nov. 2020.
27. E. Jonas, Q. Pu, S. Venkataraman, and I. Stoica, "Sprocket: A Serverless Video Processing Framework," in *Proc. ACM SoCC*, Santa Clara, CA, USA, Oct. 2017, pp. 470-484.
28. J. S. Ward and A. Barker, "FaaSdom: A Benchmark Suite for Serverless Computing," in *Proc. IEEE BigData*, Seattle, WA, USA, Dec. 2018, pp. 1-12.
29. B. Rashidi et al., "Application Execution on Serverless Edge Cloud: A Review," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 512-534, 1st Quart. 2020.

30. A. Ghosh, R. S. Mendiratta, and S. Patnaik, "Serverless Computing for Real-time Data Processing," in *Proc. IEEE ICCCI*, Chennai, India, Feb. 2020, pp. 1-5.