

REVIEW

Open Access

# Survey on serverless computing

Hassan B. Hassan<sup>1</sup>, Saman A. Barakat<sup>2</sup> and Qusay I. Sarhan<sup>2\*</sup>



## Abstract

Serverless computing has gained importance over the last decade as an exciting new field, owing to its large influence in reducing costs, decreasing latency, improving scalability, and eliminating server-side management, to name a few. However, to date there is a lack of in-depth survey that would help developers and researchers better understand the significance of serverless computing in different contexts. Thus, it is essential to present research evidence that has been published in this area. In this systematic survey, 275 research papers that examined serverless computing from well-known literature databases were extensively reviewed to extract useful data. Then, the obtained data were analyzed to answer several research questions regarding state-of-the-art contributions of serverless computing, its concepts, its platforms, its usage, etc. We moreover discuss the challenges that serverless computing faces nowadays and how future research could enable its implementation and usage.

**Keywords:** Survey, Cloud computing, Serverless computing, Serverless platforms, Serverless benefits, Serverless challenges

## Introduction

Cloud computing emerged after the appearance of virtualization in software and hardware infrastructures; hence cloud providers increasingly adopted it to offer their services to customers [1, 2]. Customers can access these cloud services via the Internet. Software developers have been using cloud technologies in their software solutions owing to their benefits including scalability, availability, and flexibility [3].

In general, cloud computing is divided into three main categories based on the provision of services, which are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). In the SaaS category, cloud providers offer different types of software as services to the users. For example, Google provides many applications as a service (e.g., Gmail, Google docs, Google sheets, and Google forms). In this type of cloud, the user is not responsible for the services development, deployment, and management. The user here only uses them without worrying about their settings, configurations, etc.

Meanwhile, in the PaaS, cloud companies provide services such as network access, storage, servers, and operating systems to be purchased by developers. The developers access these services to deploy, run, and manage their applications. In this kind of cloud, the developer is responsible for the deployment and management (settings and configurations) of their software to ensure that the application is running, while they do not control the services. Finally, in the IaaS category, the cloud consumers control and manage services such as network access, servers, operating systems, and storage.

Managing cloud services is not an easy task at all. The authors in [4] have addressed several challenges while managing a cloud environment by a user such as availability, load balancing, auto-scaling, security, monitoring, etc. For example, the cloud user has to ensure the availability of the services in which if a single machine failure occurs, it does not affect the whole services. Also, he/she has to consider distributing copies of the services geographically to protect them when disasters happen. Another challenge is load balancing. In this case, the cloud user has to ensure that requests to the services are balanced to utilize all resources efficiently.

\*Correspondence: [qusay.sarhan@uod.ac](mailto:qusay.sarhan@uod.ac)

<sup>2</sup>Software Engineering and Embedded Systems (SEES) Research Group, Department of Computer Science, College of Science, University of Duhok, Duhok, Kurdistan Region, Iraq

Full list of author information is available at the end of the article



© The Author(s). 2021 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

These challenges have led to introduce another cloud computing model, which is called serverless cloud computing [4]. Serverless cloud computing offers backend as a service (BaaS) and function as a service (FaaS), as shown in Fig. 1. The BaaS includes services like storage, messaging, user management, etc. While, the FaaS enables developers to deploy and execute their code on computing platforms. The FaaS relies on the services provided by the BaaS such as a database, messaging, user authentication, etc. The FaaS is considered as the most dominant model of serverless, and it is also known as “event-driven functions” [5, 6].

Serverless cloud model was for the first time introduced by Amazon Lambda in 2014, after which cloud companies like Google and Microsoft adopted it in 2016. Serverless cloud computing adds an additional abstraction layer to the existing cloud computing paradigms, while it abstracts away the server-side management from the developers [7]. Serverless model lets the developers focus on the application logic rather than the server-side management and configurations. For example, the developers deploy their applications to the serverless cloud as functions see Fig. 1. Then, the cloud provider takes responsibility for managing, scaling, and providing different resources to ensure the smooth running of these functions [8, 9].

However, FaaS and the term “serverless” could be used interchangeably, as the FaaS platform automatically configures and maintains the execution context of functions and connects them to cloud services without requiring server provision by developers [10, 11]. We refer to the FaaS when we use the term serverless computing.

Serverless cloud computing has many good characteristics [12, 13], one of which is scalability. Scaling could be vertical or horizontal; vertical scaling adds or removes cores from the running container, while horizontal scaling creates new containers or eliminates running ones without affecting the current resource allocations [14]. In serverless computing, the applications automatically scale up and down on demand, and the developer does not have

to concern themselves about the scaling issues. For example, when an application runs on a serverless cloud, it will scale up automatically when the application requests increase. Another characteristic of serverless computing is the payment per resource usage. This paradigm of cloud computing charges developers based on the actual resource usage. For example, deploying an application will not cost the developer in the case where the application is idle, and the serverless provider will only charge whenever the application has started using resources.

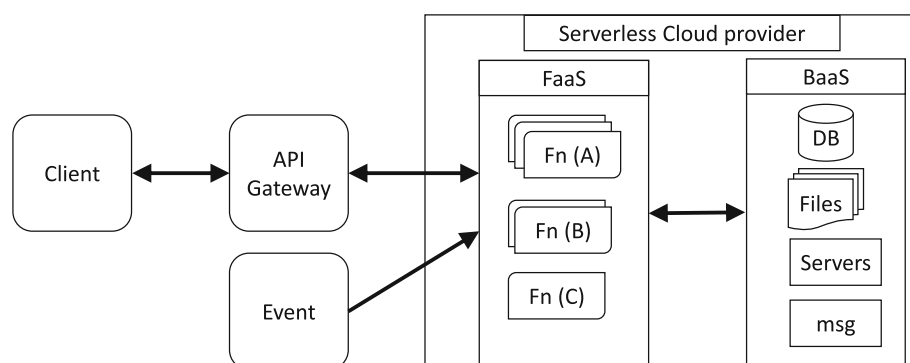
However, any new technology will face numerous technical and operational issues and obstacles at the beginning. Since the recent introduction of serverless cloud computing, several drawbacks have been identified [7]. Serverless cloud computing lacks tools that help managing and monitoring serverless applications. Moreover, it might comprise security concerns. Further, the serverless providers have a vendor lock-in problem. Nevertheless, serverless cloud computing has gained positive attention in the industry, despite that it has not been studied extensively in academic research [7].

Therefore, the aim of this research is to answer some crucial research questions related to serverless cloud computing and thereby help researchers as well as developers to better understand serverless cloud computing and contribute to its development.

The rest of this paper is structured as follows: “[Related works](#)” section presents the related works for this study. “[Research methodology](#)” section describes in detail the research methodology used to conduct this survey study. “[Results](#)” section presents the results and outcomes of the study. “[Threats to validity](#)” section presents the threats to validity of this study. Finally, the conclusions of the study are provided in “[Conclusions](#)” section.

## Related works

The most relevant studies published on the topic are briefly presented here. The authors in [15] and [16] discussed some important background to the origin and



**Fig. 1** Serverless architecture

evolution of serverless computing and the long road that serverless computing has taken over the years. The authors in [9] thoroughly discussed the true meaning of serverless architectures and how they are changing the way in which applications are built, deployed, and distributed.

Numerous studies focused on technical interpretations of serverless computing, while other more recent research suggested various benefits that it brings to developers. Nowadays, this type of computing is being used in several ways. In an empirical study, the authors in [17] aimed to investigate the development practices of serverless computing in the industry. They concluded that for developers, it remains a barrier to adopt the right mindset to best utilize the tools inherent to serverless architecture. More documentation and easier access to such resources would help developers to embrace the possibilities that serverless computing has to offer.

The concept of serverless computing within the scope of the IT industry has the great potential of progressively increasing its capabilities to involve a wider set of domains. Thus, the implementation of serverless computing is not restricted only to the enhancement of infrastructure, and it can be employed for many different purposes, e.g., serverless messaging, neural network training [18], video processing [19], and big data [20]. Undeniably, their contributions are valuable to the general public and researchers in the field, as it is of primarily importance to comprehend how this technology works.

However, it is presently crucial to provide more than only theories and concepts: it is time to weigh the benefits and drawbacks of serverless computing and to analyze how far the field has progressed, to assess what remains to be done and improved. As an example, the authors in [21] discussed some possible new abstraction levels to reduce processing limitations. The authors in [22] discussed the results from an open-source framework to achieve on-premises serverless computing that can process big workloads with a scalable and sensible usage of resources. We can infer from these related publications that researchers everywhere are working to determine how to best exploit the potential that serverless computing frameworks could introduce to software development.

In [23], the authors described how serverless computing is becoming the next step in the evolution of cloud computing and its platforms. In our paper, we focus on the ongoing challenges, benefits, and drawbacks of using it.

The authors in [24] have conducted a systematic exploration of serverless computing-related research papers. As they mentioned, their work is not a survey, but it is a supporting source for future research papers. They proposed an open dataset for serverless computing papers. The dataset includes 60 papers for the period (2016–July 2018). Also, they have analyzed the dataset according to

bibliometric, content, technology, and produced statistics about each section. In contrast, our paper aims to conduct a systematic survey. In this survey, we try to find answers to several critical questions related to serverless computing. In addition to that, our study covered the duration (2016–2020) and thus 275 papers have been considered.

The authors in [25] mainly focused on scheduling tasks in the cloud. They described the various techniques in scheduling workflows to reduce the execution time, cost, or both. Moreover, they proposed a hybrid method by both FaaS and IaaS. The small tasks could be executed remotely using the FaaS, which reduces the execution cost; hence, the number of virtual machines will be decreased as well. Therefore, the whole focus would be on the long-running tasks on IaaS.

The authors in [26] covered only 24 research papers during 2017–2019. In their paper, they considered both the white and grey literatures. Besides, they identified 32 characteristics of serverless and the possible issues related to them, only eight of them were stated by both white and grey literatures while the remaining are from grey literature only. All the characteristics are explained and presented briefly. In our paper, 275 research papers from 2016–2020 have been covered and more research questions have been answered. Besides, a well-defined systematic literature study process has been employed. Thus, the grey literature has been excluded in our paper and, our results are reproducible compared to their results.

The authors in [27] mainly concentrated on difficulties and gaps in data-centric and distributed computing using FaaS. Additionally, they evaluated the severity of these challenges via taking three case studies from big data and distributed computing settings: model training, low-latency prediction serving using the batch and, distributed computing. While our paper is a broad and comprehensive study on FaaS, 275 research papers are taken from the white literature during 2016–2020.

The paper [28] presented only four use cases of FaaS: event-triggered computing, video broadcasting, Internet of Things (IoT) data processing, and shared delivery system. Additionally, the paper only compared three platforms namely, Amazon web services (AWS) Lambda, Google Cloud Function, and Microsoft Azure Function. On the other hand, our paper presents a comprehensive study about FaaS. We identified in detail eight use cases: chatbot, information retrieval, file processing, smart grid, security, networks and, mobile and IoT. Moreover, our paper compared ten FaaS platforms namely, AWS Lambda, Apache OpenWhisk, Microsoft Azure functions, Google Cloud functions, OpenLambda, IBM Cloud functions, OpenFaaS, Knative, FunctionStage, Huawei Cloud, and Nuclio.

The authors in [29] covered only 15 papers during 2016–2018. They took both the white and grey literatures

into account. On the other hand, our paper includes 275 research papers published in the period 2016–2020; they are taken from the white literature only. Moreover, our paper has formulated and answered eight clear and well-defined research questions.

The authors in [30] focused on the FaaS performance evaluation and their publication trends during 2016–2019. They identified the most commonly evaluated FaaS platforms. Additionally, they evaluated the performance features for benchmark types, micro-benchmarks, and common features across FaaS platforms. Moreover, they presented the most common platform configurations in FaaS, namely language runtimes, function triggers, and external services. This paper presents a survey of the most important and state of the art aspects of FaaS. Besides, comprehensive theoretical aspects of FaaS are covered taking from the white literature during 2016–2020.

The authors in [11] have conducted a systematic mapping study on serverless cloud computing. The main aim of their study is to concentrate on FaaS engineering. They raised two main concerns: (a) identifying publication research that considers developing or modifying serverless platforms and tools. (b) identifying the challenges and drivers related to these publications. On the other hand, our study extends the challenges and issues related to serverless computing. Moreover, we provide more details about serverless computing platforms and the use of these platforms in the literature. Also, it provides a detailed comparison among the most widely used serverless platforms. Besides, it addresses more aspects of serverless cloud computing such as application areas of serverless computing, future directions of serverless computing, etc.

The authors in [7] provided useful observations about serverless computing, its architecture, and use cases. Also, they discussed the challenges and benefits of moving forward from monolithic applications and the differences between traditional cloud services and serverless computing. Our work has extended the details of their work regarding the benefits and drawbacks of using serverless computing. It has also included more use cases and workloads to deepen the findings of previous studies.

The authors in [4] presented a technical report on serverless computing. They covered the serverless emergence with its limitations, including limited storage for fine-grained tasks, lack of coordination among functions, inadequate performance for standard communication patterns, and functions' performance. Also, they compared AWS serverful with AWS serverless. Moreover, they also explained the challenges of architecture, networking, security, and abstractions of serverless computing. They identified five application areas including, video encoding in real-time, MapReduce, linear algebra, machine learning training, and databases. While our

paper has covered 275 research papers from 2016–2020 forming a well-defined systematic literature study. We also identified 21 serverless challenges and issues. Besides, we compared serverless with the traditional cloud computing paradigm. We identified more application areas including, chatbot, information retrieval, file processing, smart grid, security, networks, IoT, and edge computing.

The authors in [31] presented a white paper based on published research papers during 2015–2017. They outlined the serverless definition alongside its advantages and disadvantages. Also, they classified serverless use-cases into six categories, namely, backends, web applications, chatbots, big data, IT automation, and Amazon Alexa. Moreover, they addressed a few research questions including, the need for the stateless feature in serverless, whether serverless could execute long-running tasks, programming models, serverless standards, and the importance of serverless for scientific research. While our paper is a comprehensive study on FaaS; we covered 275 research papers which are taken from the grey literature during 2016–2020. In our paper, eight application areas have been identified as mentioned earlier. We have identified and answered ten research questions that cover many aspects of the topic in detail compared to the aforementioned study.

We are in fact addressing with this paper ten important research questions about the topic, potentially making it a more complete guide to the development and use of serverless computing. Our work contributes to the analysis of the serverless paradigm in the context of similar applications and how could they better fit specific computing needs. Moreover, information about the current state of serverless platforms, tools, and frameworks has been updated for this survey. This due to the importance of the topic and its potential to change how both the industry and academia have managed the deployment of cloud applications until now. Updated information about the area could benefit future studies focused on the serverless computing paradigm as they make researchers aware of the latest resources and opportunities in the area.

## Research methodology

### Research questions

In this study, a number of research questions (RQs) have been identified and answered. Each RQ addresses a particular aspect of serverless computing as follows.

- RQ1. What is the number and distribution of studies published on serverless computing in the period (2016–2020)?
- RQ2. Which researchers, organizations, and countries are active in serverless computing research?
- RQ3. What are the differences between serverless computing and traditional cloud computing?



- RQ4. What are the benefits of using serverless computing?
- RQ5. What are the most used software platforms that enable serverless computing in the literature?
- RQ6. What are the application areas of serverless computing in the literature?
- RQ7. What are the challenges and issues of using serverless computing?
- RQ8. What tools are available for serverless computing? (serverless tools)
- RQ9. What are the available research approaches to analyze the migration of monolithic applications to serverless computing?
- RQ10. What are the potential future directions of research on serverless computing?

### Search strategy

#### Literature sources

In this study, five standard online databases have been selected as sources that index the literature of software engineering and computer science. These sources are presented in Table 1.

#### Search string

To find the publications relevant to this study, the following extensive search string has been applied on the database sources of literature:

*(serverless OR FaaS OR “function as a service” OR “function-as-a-service”) AND (computing OR paradigm OR architecture OR model OR application OR function OR service OR platform OR programming)*

To obtain the best publication list, a generic search string is created. It contains serverless cloud computing-related keywords. The string with duration (2016 - 2020) have been applied to all libraries. Because the Springer Link library covers many fields, the result of search was greater than other libraries. This because the keyword FaaS is used in many research areas for different purposes. For instance, fish as a service (FaaS) and FPGA as a fervice (FaaS). Therefore, we used Computer Science subject filter with Springer Link, ScienceDirect, and Scopus to reduce the number of incorrect papers. The results of the initial search are shown in Fig. 2. Additionally, some

inaccurate results have been obtained due to the partial similarity to FaaS, such as the federal aviation administration (FAA). The results of the initial search were 5,021 papers in total.

After obtaining the initial list of publications, some filters have been applied to reduce the number of incorrect results based on their relation to the serverless computing and FaaS topics. Most of the papers have been analyzed based on the title and abstract. However, when we were unable to make a decision based on the title and the abstract, we read the content of the paper to ascertain whether to include or exclude. As a result, the list of papers which are related to serverless computing has been decreased to 549 papers.

After filtering the papers based on the title and abstract, we merged all the papers that were relevant to serverless cloud computing, which was 549 papers into a single dataset. Then we removed the duplicated papers based on the combination of a title, author names, publication year, and venue. Thus, the number of publications has been reduced to 489 papers.

Then, the publications have been selected based on the content of the paper and based on a set of inclusion/exclusion criteria (see the following section) that have been selected carefully. Eventually, we could obtain 254 papers that are related to serverless cloud computing. In the next step, we applied backward snowballing to increase the set of relevant papers to serverless cloud computing. In this phase, we could add 21 more papers to our list of papers. As a result, the total numbers of relevant papers become 275 papers. The list of these papers and its meta-data have been published in Zenodo website as a dataset [32].

#### Paper inclusion/exclusion criteria

To decide whether a publication is relevant to the scope of this research, a set of inclusion and exclusion criteria have been established and employed as follows:

##### Inclusion criteria:

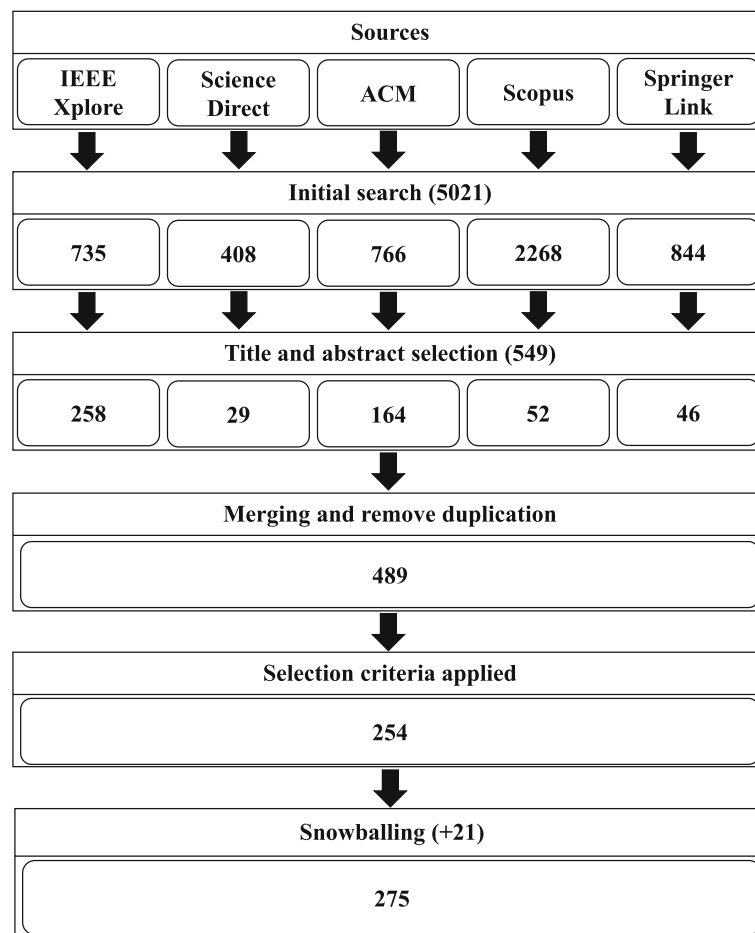
- Publications in the field of software engineering and computer science.
- Publications published online from 2016 – 2020.
- Publications related directly to serverless computing.

##### Exclusion criteria:

- Publications not published in English.
- Publications without accessible full text.
- Publications not formally peer reviewed (e.g., gray literature).
- Publications not published electronically.
- Publications that are duplicates of other previous publications.

**Table 1** Database sources used to explore the literature

Source	URL
IEEE Xplore	<a href="http://ieeexplore.ieee.org">http://ieeexplore.ieee.org</a>
Elsevier ScienceDirect	<a href="http://sciencedirect.com">http://sciencedirect.com</a>
ACM Digital Library	<a href="http://portal.acm.org">http://portal.acm.org</a>
Scopus	<a href="http://scopus.com">http://scopus.com</a>
SpringerLink	<a href="http://springerlink.com">http://springerlink.com</a>



**Fig. 2** Results of papers selection process

## Results

The selected publications were carefully read to answer the raised RQs. Here, a short title is used to represent each RQ. The following subsections present and discuss the results based on each RQ.

### Distribution of publications (RQ1)

#### Publication frequency

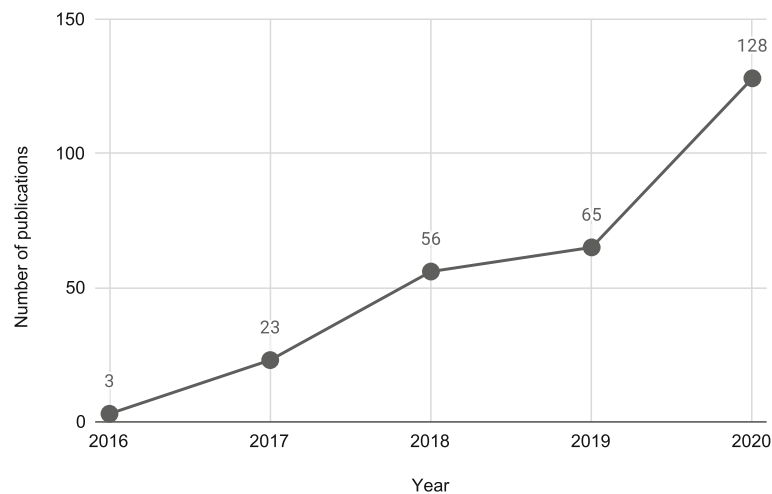
All the selected papers of this study were analyzed to determine their frequency and evolution. Figure 3 shows the results of this analysis. The results show that the average number of publications per year is approximately 55 papers.

Serverless computing has trended a significant engagement over the past two years. This boost has been caused by industry, academia, and developers for several reasons. The first important reason is the attractive engagement opportunities that serverless offers cloud providers. Serverless nature equipped cloud providers with more convenient and efficient methods to manage and utilize idle computing resources. Another reason is that

the billing is only on the basis of function execution time and resource allocation. Also, the developers are not required to be aware of the underlying infrastructure and workflows. Hence, this attracts cloud providers and businesses to migrate and support serverless alongside many directions. At the same time, researchers are paying more attention to serverless as it is becoming the future paradigm for cloud computing. Moreover, current challenges and limits in serverless computing draw attention to more academics to address the issues and enhance the currently available features. For the aforementioned reasons, developers and customers are well encouraged and satisfied to select serverless computing for developing applications and services.

#### Publication venue

The distribution of the selected papers in various publication venues is shown in Fig. 4. The percentages of publications in conference papers, workshop papers, symposium papers, and journal papers are approximately 62%, 11%, 14%, and 13%, respectively. However, almost 13% of

**Fig. 3** Published papers per year

the studies have been published in journals, which indicates the immaturity of research in serverless computing [33, 34]. It is worth mentioning that some conference papers were published as book chapters. Thus, the original venues, which are conferences, of such papers were considered.

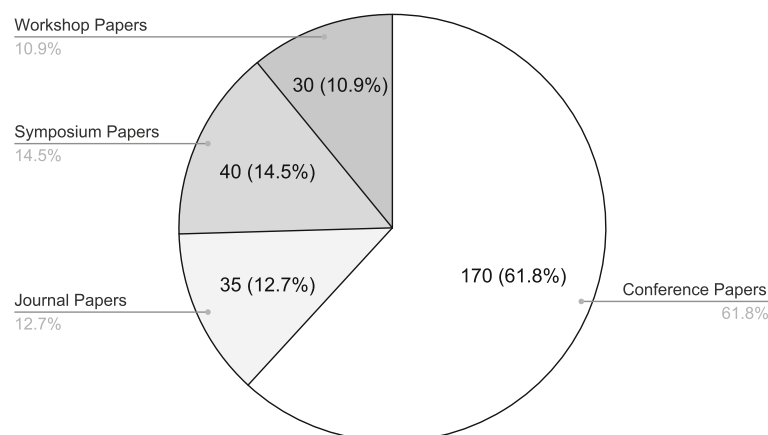
Following the interpretation of publications, the most productive and primary journals, symposiums, conferences, and workshops venues related to serverless computing can be clarified. Due to their long names, abbreviations are used in this paper. The active journals are shown in Fig. 5 and their full names can be found in Table 2. It can be observed from the figure that the top and vital three journals are “FGCS”, “IoT”, and “JSS”. Also, it can be noticed that the top three journals contain almost 34% of the published journal papers, while the others own approximately 66%.

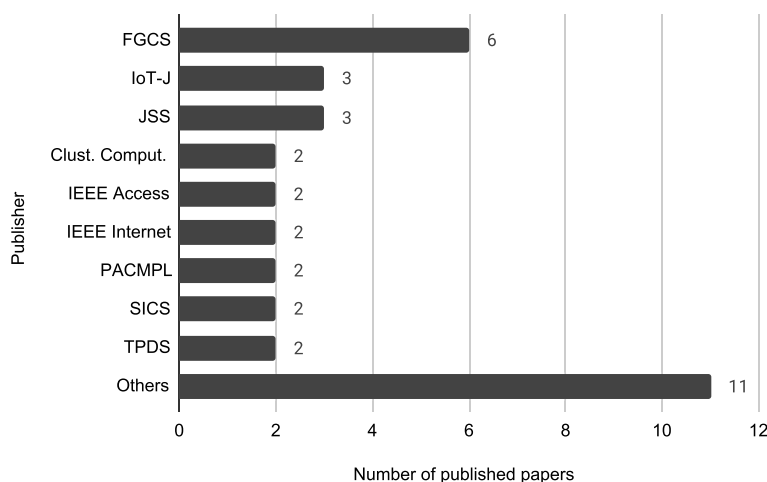
The active conferences are shown in Fig. 6 and their full names are presented in Table 3. The “WOSC”,

“Cloud”, “UCC”, “SoCC”, and “Middleware” are considered the most active conferences that hold approximately 28% of the published conference papers. By including other conferences with three published papers or more, then approximately 23% of the conference papers are published in annual conferences. The majority (almost 49%) of the conference papers were published at individual conferences, which are denoted as “Others” in Fig. 6.

#### Active researchers (RQ2)

Serverless computing is a vital research area through the contribution of several scholars. Yet, the researchers are counted active if they contributed to more than two research studies, as presented in Fig. 7. The figure shows that the top six active researchers are “Pedro García López”, “Erwin Van Eyk”, “Alexandru Iosup”, “Marc Sánchez-Artigas”, “Sebastian Werner”, and “Wes Lloyd”. Table 4 presents the active nations, research institutions,

**Fig. 4** Published papers ratio per each venue



**Fig. 5** Published papers vs. journal name

researchers, references to the published papers, and the total number of publications.

The active nations in the number of papers are obtained from the information presented in Table 4 by extracting the institutional affiliation of the authors and co-authors. An overview of the most active nations and the total number of publications is shown in Fig. 8. It is observable that the United States and Germany are the largest contributors to papers published on serverless computing with 104 and 39 published papers, respectively.

#### Serverless computing vs. traditional cloud

computing (RQ3) There are several differences between serverless and traditional cloud computing. In the traditional cloud architecture, the server acts as a monolithic system containing all business logic. Meanwhile, the serverless architecture is modeled into smaller, event-driven, and stateless ‘triggers’ (events) and ‘actions’ (functions) [175]. Each component handles different pieces of data and runs independently [176]. Spreading business

logic into smaller functions increase the development efficiency [77, 177] and also decreases the chance of a single point of failure [77]. On the other hand, the component dependency within monolithic applications affects the availability of other services adversely.

In a serverless architecture, the developers are unable to take control of listening to the TCP socket, managing load balancers, maintenance or configuration of the server, as well as provisioning and resource allocation. Therefore, there is no need for system administrators; the developers only focus on handling client requests and paying attention to deliver valuable services [8].

Serverless computing also differs from monolithic computing as the functions have shorter life cycles.

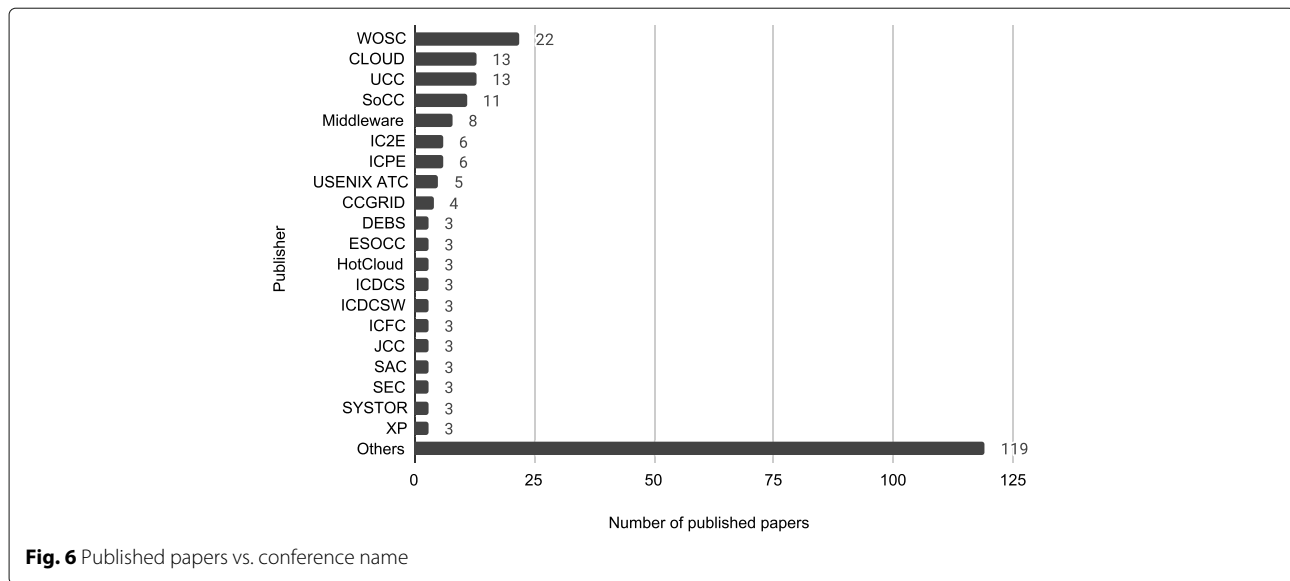
The traditional monitoring and debugging tools that are used in monolithic applications are not included in the serverless architecture; the developers are compelled to use built-in tools for debugging and monitoring. The computing power is no longer a concern for the developers in the serverless paradigm, as it could scale horizontally almost indefinitely [178, 179]. Meanwhile, in the client-server architecture, it usually requires dedicating two server instances; the primary instance and a second in case if the former fails. This leads to higher costs in the monolith paradigm. Serverless architecture could be more economical for unsteady load conditions while the server-based is more suitable for steady loads [152]. As serverless applications scale up and down according to the requests, thus, unlike the traditional systems, it is unnecessary to keep the sessions in the memory [8]. Hence, it is difficult to keep track across requests.

FaaS boosts the security level as cloud providers continuously update their infrastructure with the latest security patches; this also removes the security burden on developers [17]. Directly accessing the backend resources in the traditional model is considered a critical security issue.

**Table 2** List of active journals

Acronym	Journal Full Name
FGCS	Future Generation Computer Systems
IoT-J	Internet of Things Journal
JSS	Journal of Systems and Software
Clust. Comput.	Cluster Computing
IEEE Access	IEEE Access
IEEE Internet Comput.	IEEE Internet Computing
PACMPL	Proceedings of the ACM on Programming Languages
SICS	Software-Intensive Cyber-Physical Systems
TPDS	IEEE Transactions on Parallel and Distributed Systems



**Table 3** List of active conferences

Acronym	Conference Full Name
WOSC	International Workshop on Serverless Computing (WOSC)
Cloud	International Conference on Cloud Computing (Cloud)
UCC	International Conference on Utility and Cloud Computing (UCC)
SoCC	Symposium on Cloud Computing (SoCC)
Middleware	International Middleware Conference (Middleware)
IC2E	International Conference on Cloud Engineering (IC2E)
ICPE	International Conference on Performance Engineering (ICPE)
USENIX ATC	USENIX Annual Technical Conference (USENIX ATC)
CCGRID	International Symposium on Cluster, Cloud and Grid Computing (CCGRID)
DEBS	International Conference on Distributed and Event-based Systems (DEBS)
ESOC	European Conference on Service-Oriented and Cloud Computing (ESOC)
HotCloud	Conference on Hot Topics in Cloud Computing (HotCloud)
ICDCS	International Conference on Distributed Computing Systems (ICDCS)
ICDCSW	International Conference on Distributed Computing Systems Workshops (ICDCSW)
ICFC	International Conference on Fog Computing (ICFC)
JCC	International Conference on Joint Cloud Computing (JCC)
SAC	Symposium on Applied Computing (SAC)
SEC	Symposium on Edge Computing (SEC)
SYSTOR	International Systems and Storage Conference (SYSTOR)
XP	International Conference on Agile Software Development (XP)

Thus, any requests from the clients and internal functions in the serverless environment must go through a distributed request-level authorization mechanism that strengthens the security level [8]. Additionally, denial of service (DoS) attacks are controlled, as it is more difficult to attack distributed servers than a single server [175]. However, some security concerns remain due to the third-party API usage [9]. Besides, there is a lack of tools to identify vulnerabilities and access control risks. Table 5 summarizes the aforementioned differences.

#### Benefits of serverless computing (RQ4)

Serverless computing offers numerous benefits to its users, and Table 6 presents papers that states these benefits. This section summarizes those benefits as follows:

##### Cost effective

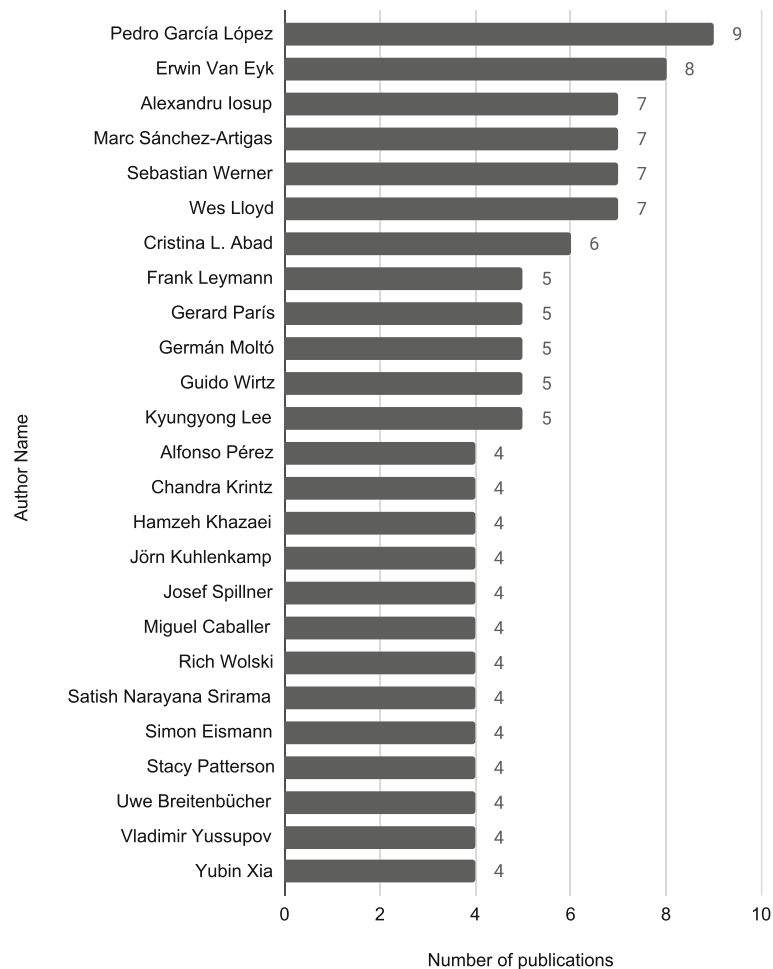
Serverless applications are abstracted from server infrastructure, which results in cost-based services depending on usage [180]. For example, applications run whenever a user makes a request to a service within the application. The cloud vendors charge only for the used space, and there is no cost while their applications are in an idle state.

##### Scalability

Serverless reasonably solved the resource allocation problem [191]. Therefore, developers do not have to concern themselves with the application scalability, because the application will scale automatically whenever user application requests are increased. If there are numerous requests to a function within the application, the serverless providers will start servers to handle these requests.

##### Server-side management

In serverless computing, developers do not need to concern themselves with the server-side and its management.



**Fig. 7** Active researchers based on the published papers

Serverless cloud providers take care of managing and maintaining the hardware and software required to deploy applications. In addition to that, they handle all administration operations to let developers focus on different kinds of resources such as central processing unit (CPU), memory, and storage.

#### **Easy to deploy**

Serverless applications are easy to deploy. For example, to deploy an application, developers only need to upload some functions and release a new product. The serverless will take care of deployment management and infrastructure related concerns such as server provisioning and scaling.

#### **Decrease latency**

Serverless applications are not hosted on a specific server; the code can run from any server in any location. Therefore, cloud vendors can run the application on servers near the end user's location. This reduces latency, because end

user requests do not have to travel across the Internet to access the original server.

#### **Serverless platforms in the literature (RQ5)**

The software platforms are generally implemented to deal with resources from several clouds and ensure proper running of client applications. The heterogeneous nature of the cloud providers' infrastructure (hardware and operating systems) led to the necessity to direct the developers' focus to the functional part, rather than the underlying infrastructure [199].

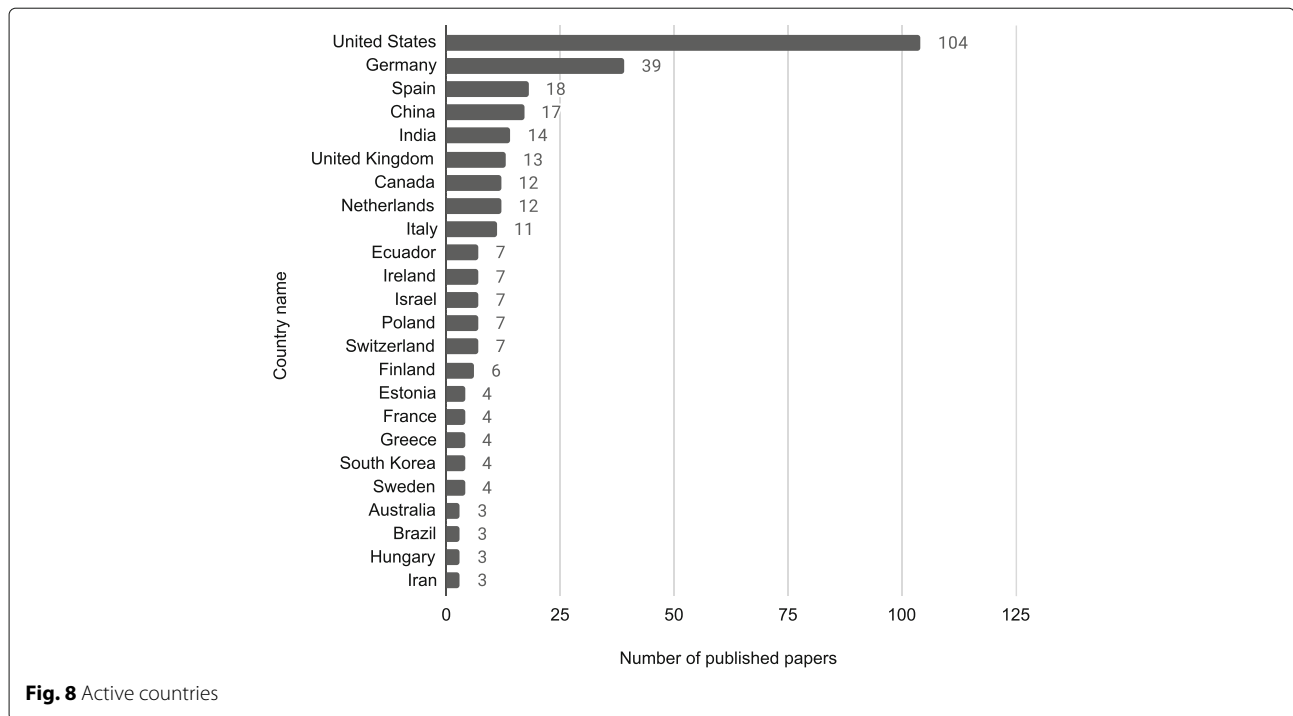
With the emergence of the first serverless platform, AWS Lambda by Amazon in 2014 [8], cloud computing has evolved to a new generation referred to as serverless computing. However, serverless was not a brand-new paradigm; it emerged after the advancements in adopting virtual machines and container technologies [120]. By 2016, other competitors, namely Google, Microsoft, and IBM followed the trend. Several commercial and open-source platforms offer serverless computing. The well-

**Table 4** Organizations, and researchers active in researching serverless computing

Organization	Author(s)	Published Paper	Total
IBM Research, United States	Alaa Youssef, Aleksander Slominski, Ali Anwar, Ali Kanso, Animesh Trivedi, Bruce Slawson, Canturk Isci, Dan Williams, Dimitrios Skourtis, Eric Rozner, Erik Wittern, Ioana Baldini, Jonas Pfefferle, Kerry Shih-Ping Chang, Lionel Villard, Lukas Rupprecht, Michael Maximilien, Nick Mitchell, Nilton Bila, Nima Kaviani, Nirmal K Mukhi, Olivier Tardieu, Paolo Dettori, Patrick Stuedi, Paul Castro, Pedro Garc�a L�pez, Perry Cheng, Philippe Suter, Prabhakar Kudva, Ricardo Koller, Rodric Rabbah, Shripad Nadgowda, Sriji Prabhhu, Stephen Fink, Vasily Tarasov, Vatche Ishakian, Vinod Muthusamy, Waldemar Hummer	[17, 18, 21, 35–54]	23
Technische Universit�t Berlin, Germany	Jorn Kuhlenkamp, Sebastian Werner, Maria C. Borges, Stefan Tai, Johannes M�ller, Richard Girke, J�rn Kuhlenkamp, Dominik Ernst, Frank Pallas, Niklas Anders, Nebi Muc�j, Olesia Tsaplina, Christian Schmidt, Kann Yildirim, Tobias Pfandzelter, David Bermbach, Ahmet-Serdar Karakaya, Simon Buchholz, Setareh Maghsudi, Jonathan Hasenburger	[13, 20, 55–62]	10
Universitat Rovira i Virgili, Spain	Pedro Garc�a L�pez, Marc S�nchez-Artigas, Gerard Par�, Daniel Barcelona Pons, �lvaro Ruiz, David Arroyo Pinto, Josep Samp�, Amanda G�mez-G�mez, Mariano Ezequiel Mirabelli, Germ�n T. Eizaguirre, Aitor Arjona	[45, 46, 49–51, 63–66]	9
University of California, Berkeley, United States	Andrew Zhang, Benjamin Recht, Chenggang Wu, Dominic Carrano, Ion Stoica, Jeffrey Ichnowski, Joao Carreira, Joseph E. Gonzalez, Joseph M. Hellerstein, Kannan Ramchandran, Karl Krauth, Ken Goldberg, Qifan Pu, Randy Katz, Samuel Paradis, Saurav Chhatrapati, Scott Shenker, Thomas Courtade, Vaishaal Shankar, Vikram Sreekanti, Vipul Gupta, Vivian Fang, Wen Zhang, Yaoqing Yang	[67–74]	8
University of Washington, Tacoma, United States	Baojia Zhang, David Foster, David Perez, Derek Chen, Dimitar Kumanov, Eyhab Al-Masri, Hanfei Yu, Ibrahim Diabate, Ka Yee Yeung, Ling-Hong Hung, Ming Hoi Lam, Minh Vu, Mohammadbagher Fotouhi, Rashad Hatchett, Richa Jain, Robert Cordingley, Shruti Ramesh, Swetha Chinthalapati, Swetha Reddy Nathala, Varik Hoang, Wen Shu, Wes Lloyd, Xingzhi Niu, Zohreh Sadeghi	[75–82]	8
Vrije Universiteit Amsterdam, Netherlands	Lucian Toader, Alexandru Uta, Ahmed Musaafir, Alexandru Iosup, Laurens Versluis, Erwin Van Eyk	[16, 83–89]	8
University of Wisconsin-Madison, United States	Aditya Akella, Andrea C. Arpaci-Dusseau, Arjun Singhvi, Dennis Zhou, Edward Oakes, Kevin Houck, Leon Yang, Liang Wang, Michael Swift, Remzi H. Arpaci-Dusseau, Scott Hendrickson, Shivaram Venkataraman, Stephen Sturdevant, Tyler Harter, Venkateshwaran Venkataramani	[71, 74, 90–94]	7
Delft University of Technology, Netherlands	Erwin Van Eyk, Alexandru Iosup, Sacheendra Talluri	[16, 84–86, 89, 95]	6
Escuela Superior Politecnica del Litoral, Ecuador	Gabriel Aumala, Edwin F. Boza, Luis Ortiz-Avil�s, Gustavo Totoy, Cristina L. Abad, M�nica Villavicencio, Stephany Quimba	[2, 85–87, 95, 96]	6
Stanford University, United States	Ana Klimovic, Yawen Wang, Christos Kozyrakis, Sadjad Fouladi, Keith Winstein, Kostis Kaffes, Neeraja J. Yadwadkar, Sean Choi, Muhammad Shahbaz, Balaji Prabhakar, Mendel Rosenblum, Kailas Vodrahalli	[43, 53, 71, 97–99]	6
University of Bamberg, Germany	Johannes Manner, Martin Endre�, Tobias Heckel, Guido Wirtz, Stefan Kolb, Stefan Winzinger, Marcel Gro�mann, Christos Ioannidis, Duy Thanh Le, Mike Prechtl, Robin Lichtenth�ler	[1, 100–104]	6
Imperial College London, United Kingdom	Robert Chatley, Alim Ul Gias, Giuliano Casale, Alessandra Russo, Lulai Zhu, Jessica Vandebon, Jos� G. F. Coutinho, Wayne Luk	[8, 105–108]	5
Kookmin University, South Korea	Jeongchul Kim, Kyungyong Lee, Jungae Park, Hyunjune Kim, Jaeghang Choi	[109–113]	5
Shanghai Jiao Tong University, China	Mingyu Wu, Zeyu Mi, Yubin Xia, Zijun Li, Quan Chen, Shuai Xue, Minyi Guo, Dong Du, Tianyi Yu, Binyu Zang, Haibo Chen, Qingyuan Liu, Ziqian Lu, Jinfeng Liu, Zheng Huang, Zhichao Hua	[114–118]	5
Universitat Polit�cnica de Val�ncia, Spain	Alfonso P�rez, Germ�n Molt�, Miguel Caballer, Amanda Calatrava, Sebasti�n Risco, Diana M. Naranjo, V. Gim�nez-Alventosa, Carlos De Alfonso, Ignacio Blanquer	[22, 119–122]	5

**Table 4** Organizations, and researchers active in researching serverless computing (*Continued*)

Organization	Author(s)	Published Paper	Total
University of Alberta, Canada	Di Niu, Sara Ghaemi, Alireza Goli, Omid Hajihassani, Omid Ardakanian, Moe Rashidi, Tyler Dauphinee, Nima Mahmoudi	[123–127]	5
University of Stuttgart, Germany	Michael Wurster, Uwe Breitenbücher, Kálmán Képes, Frank Leymann, Vladimir Yussupov, Michael Hahn, Christian Müller, André van Hoorn	[11, 107, 128–130]	5
Zurich University of Applied Sciences, Switzerland	Josef Spillner, Seán Murphy, William Martini, Bill Bosshard	[17, 131–134]	5
IBM Research, Israel	Alex Glikson, Gil Vernik	[23, 49, 65, 66]	4
Intel Corporation, United States	Michael Steiner, Ramesh G. Illikkal, Eriko Nurvitadhi, Mishali Naik, Anup Mohan, Harshad Sane, Kshitij Doshi, Saikrishna Edupuganti, Naren Nayak	[108, 135–137]	4
Microsoft Research, United States	Lan Ly, Mohammad Shahrad, Rodrigo Fonseca, Íñigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, Ricardo Bianchini, Tyler Harter, Jose M. Faleiro	[68, 75, 91, 138]	4
Platform9 Inc., United States	Erwin Van Eyk	[84, 85, 89, 95]	4
Rensselaer Polytechnic Institute, United States	Matthew Obetz, Anirban Das, Timothy Castiglia, Stacy Patterson, Ana Milanova, Shigeru Imai, Andrew Leaf, Carlos A. Varela	[139–142]	4
University of California, Santa Barbara, United States	Wei-Tsung Lin, Chandra Krintz, Rich Wolski, Michael Zhang, Fatih Bakir, Gareth George	[10, 143–145]	4
University of Illinois, Urbana-Champaign, United States	Mengting Yan, Pubali Datta, Adam Bates, Arnav Sankaran, Tarek Elgamal	[44, 146–148]	4
University of Tartu, Estonia	Chinmaya Kumar Dehury, Satish Narayana Srirama, Tek Raj Chhetri, Pelle Jakovits	[107, 149–151]	4
University of Würzburg, Germany	Johannes Grohmann, Simon Eismann, André Bauer, Norbert Schmitt, Nikolas Herbst, Samuel Kounev	[85–88]	4
York University, Canada	Hamzeh Khazaei	[124–127]	4
AGH University of Science and Technology, Poland	Maciej Malawski, Adam Gajek, Adam Zima, Bartosz Balis, Kamil Figiela, Piotr Moczurad, Maciej Pawlik, Pawel Banach	[152–154]	3
Boston University, United States	Anthony Byrne, Ayse K. Coskun, Nabeel Akhtar, Ali Raza, Ibrahim Matta, James Cadden, Thomas Unger, Yara Awad, Han Dong, Orran Krieger, Jonathan Appavoo	[48, 155, 156]	3
Budapest University of Technology and Economics, Hungary	David Balla, Markosz Maliosz, Csaba Simon, János Czentye, Balázs Sonkoly	[157–159]	3
Einstein Center Digital Future Mobile Cloud Computing Research Group, Germany	Tobias Pfandzelter, David Bermbach, Ahmet-Serdar Karakaya, Simon Buchholz, Setareh Maghsudi, Jonathan Hasenburger	[60–62]	3
George Mason University, United States	Benjamin Carver, Jingyuan Zhang, Ao Wang, Yue Cheng	[52, 54, 160]	3
George Washington University, United States	Guyue Liu, Timothy Wood, Phani Kishore Gadepalli, Sean McBride, Gregor Peach, Gabriel Parmer	[6, 161, 162]	3
Google Inc., United States	Qifan Pu, Junaid Khalid, Vadim Sukhomlinov	[71, 93, 137]	3
Indian Institute of Technology Bombay, India	Nikhila Somu, Nilanjan Daw, Umesh Bellur, Purushottam Kulkarni, S. C. Gupta	[163–165]	3
Macquarie University, Australia	Young Choon Lee	[166–168]	3
Stony Brook University, United States	Amoghavarsha Suresh, Gagan Somashekar, Anandh Varadarajan, Veerendra Ramesh Kakarla, Hima Upadhyay, Anshul Gandhi, Prabuddha Kumar, Amir Rahmati	[146, 169, 170]	3
University of Colorado Boulder, United States	Zaid Al-Ali, Sepideh Goodarzy, Ethan Hunter, Sangtae Ha, Richard Han, Eric Keller, Erika Hunhoff, Shazal Irshad, Ali Tariq, Eric Rozner, Austin Pahl, Sharat Nimmagadda, Siddharth Lanka	[21, 171, 172]	3
University of Gothenburg, Sweden	Philipp Leitner, Joel Scheuner	[17, 30, 87]	3
University of Sydney, Australia	Mohammadreza Hoseinyfarahabady, Albert Y. Zomaya, Young Ki Kim	[166–168]	3
University of Waterloo, Canada	Matt Crane, Jimmy Lin, Youngbin Kim, N. Asokan	[135, 173, 174]	3



known commercial systems are AWS Lambda, Google Cloud Functions, and Azure functions. Alternately, there are several open source platforms available including IBM Cloud Functions, and Apache OpenWhisk.

There are several criteria to help developers in selecting a serverless platform: cost, performance, supported

programming languages and model, deployment easiness, easiness in composing functions from different providers, security, and monitoring and debugging tools [184].

Table 7 presents the serverless platforms used in the considered papers of this study. It can be noted that “AWS Lambda”, “Apache OpenWhisk”, and “Azure Functions” are the most used platforms with 78, 23, and 11 published papers, respectively. However, it is worth mentioning that each platform has its own set of features and differs from others.

**Table 5** Comparison between serverless and traditional computing

Factors	Traditional cloud computing	Serverless computing
Development phase	Difficult	Easy
Automatic scalability	Unavailable	Available
Stateful applications	Easy	Difficult
Security	Complex and less secure	Easy and more secure
Functions' life cycle	Long	Short
Troubleshooting and debugging	Easy	Difficult
Server and hardware configuration and maintenance	Required	Unavailable
Failure tolerance	Less reliable	More reliable
Cost (Variable workload)	Expensive	Affordable
Cost (Stable workload)	Affordable	Expensive
Applicable user	Administrator and developer	Developer

#### The application areas of serverless computing in the literature (RQ6)

Serverless computing can be utilized in a number of application areas as follows:

##### Chatbot

A chatbot application is developed using serverless computing, which enables interaction with users via voice

**Table 6** Benefits of serverless computing

Benefits	Papers	Total
Cost effective	[16, 94, 134, 180–190]	14
Scalability	[15, 16, 74, 75, 90, 94, 153, 188, 189, 191–193]	12
Server-side management	[15, 16, 121, 181, 188, 194–197]	9
Easy to deploy	[6, 15, 46, 95, 97, 131, 189]	7
Decrease latency	[9, 95, 179, 198]	4



**Table 7** Serverless platforms usage

Platforms	Papers	Total
AWS Lambda	[3, 5, 18–20, 30, 37, 46, 55, 68, 69, 75, 76, 78–80, 83, 91, 100, 101, 106, 109, 110, 119–121, 123, 129, 130, 140, 143, 150, 152, 153, 158, 160, 166, 167, 173, 174, 176, 177, 179, 181, 183, 188, 189, 200–230]	78
Apache OpenWhisk	[35, 36, 38, 41, 44, 91, 135, 183, 188, 189, 206, 208, 231–240]	23
Azure Functions	[55, 75, 77, 100, 109, 130, 183, 204, 206, 208, 230]	11
Google Cloud Functions	[79, 109, 152, 183, 206, 208, 241]	7
OpenFaaS	[22, 122, 141, 235, 242–244]	7
OpenLambda	[90–92, 96, 245, 246]	6
IBM Cloud Functions	[65, 130, 183, 188]	4
Knative	[235, 243]	2
FunctionStage	[247]	1
Nuclio	[243]	1

or text commands. Within this application, six functionalities have been implemented, namely the Date, News, Jokes, Weather, Music Tutor, and Alarm Service. For example, a user can ask for the current date using a voice or text command. The request is routed to a required serverless action on OpenWhisk for further processing. The Date action is activated via the issued command and retrieves the current date to the user in the form of text or voice [44].

Another example is the ticketing chatbot service developed using serverless computing and natural language processing (NLP). The architecture of the system consists of three parts: (1) the node.js Webhook, which works based on hypertext transfer protocol (HTTP) POST or GET requests (2) Wit.AI, which is a NLP service (3) Ticket.com, which is a ticketing order API. For example, when a user books a flight ticket; a specific function on Webhook will be activated, which routes the user query to the Wit.AI service. Wit.AI will process the query and extract useful parameters from the request such as destination, date, and time, then send it back to Webhook. After receiving the processed query from Wit.AI; another action will be triggered and pass the processed query to Ticket.com API to retrieve flight information such as the flight number, airline name, departure time, and ticket price from several airline companies. Finally, Webhook will provide flight information to the user [44, 179, 248].

#### Information retrieval

A search engine web-based application is developed based on serverless architecture. Search engine functionalities

are implemented as Amazon lambda functions. The search engine executes all the details of retrieval processing after receiving the user query (e.g., tokenization, stop-word removal, term weighting, similarity calculation, and ranking). Then, it sends back the results to the user as documents stored in the DynamoDB database to be accessed using the web application interface [173].

#### File processing

Serverless computing can be utilized in file processing applications [119, 249]. For instance, in [119] a model for highly parallel file processing applications based on serverless architecture is proposed. This model provides users with different ways to process their files.

The first method is by using the API gateway. In this method, users submit files using the HTTP request employing the API gateway to a lambda function to process the file (e.g., medical images and video files).

The second method is by uploading/reading files to the Amazon simple storage service (Amazon S3) bucket. This method provides the user with three different ways to execute a lambda function using S3 buckets: (a) by uploading a file to S3 buckets. When the file is uploaded, S3 creates an event to invoke a lambda function; (b) by copying a file from another bucket to the bucket linked with the lambda function. This will cause the trigger of an event from S3 to invoke a lambda function as in the previous manner; (c) by specifying a bucket where the files to be processed are stored. Then, for each file found, the lambda function is invoked in parallel using an S3 bucket.

The third method is by specifying the output file. By this method, the user can set a chain of lambda functions to be invoked by S3 buckets. In this case, the user defines the input/output buckets for each of the lambda functions. Thus, the output bucket can be used as an input to another lambda function [119]. Here, serverless functions can handle different types of data (stored in files) such as sensory, textual, and biological data [200]. Also, many preprocessing operations using NLP may be applied to data files before processing, such as stemming and noise removal [78].

#### Smart grid

A MATLAB simulation scenario is created to illustrate the use of the smart grid with serverless cloud computing to control and manage electrical loads (devices). In this scenario, the Simulink tool is employed for simulation. A MATLAB program is developed to indicate the start and end of the simulated grid model via a batch file. The batch file is used to upload grid model data generated by the program to Amazon S3. Afterwards, a lambda function in the serverless side will be activated to process the uploaded data, and subsequently the result will be sent back to the batch file as a response. In return, the program will read continuously the response from the batch file

and interpret its content to manage the electrical switch (loads) [201].

Also, An electrical overload warning system is implemented in the smart grid, based on serverless architecture. The system uses some Amazon web services, including S3, lambda functions, simple notification service (SNS), and CloudWatch. S3 is used as a storage service in the system. Lambda functions constitute a computing service that executes the code of the application. CloudWatch is a monitoring tool that monitors AWS resources and applications. The SNS is a notification service that sends and receives notifications.

The main sections of this warning system consist of data collection, data acquisition, data analysis, data mining, conclusion verification, and conclusion publishing. In this architecture, the AWS Lambda is used in data analysis and data mining. AWS CloudWatch is used for data conclusion verification. The SNS is used to generate alarms. For instance, the data is uploaded to S3, and subsequently, a lambda function is activated for data analysis and data mining. After the lambda function execution, its log data is stored in CloudWatch logs. CloudWatch is used for conclusion verification. CloudWatch defines an alarm size to a specific value, upon which it compares the value of log data with a predefined alarm size to check the current state. Then, the CloudWatch uses SNS for publishing conclusions. If the receiving data is greater than the alarm size, an alarm signal will be triggered and send an email via SNS [5].

### Security

An automated threat detection system is introduced using serverless cloud computing and Kubernetes. Kubernetes is an open source system to automatically deploy and manage application containers [243, 250]. The system deals with threats (e.g., software vulnerabilities and insecure configurations) automatically based on user-defined policies. The system includes a vulnerability scanner (VS), which is a thread detection component. Whenever users deploy new application containers, the containers are registered with the VS, and a scanner agent is installed. When a thread is detected by the scanner, a notification is sent to the OpenWhisk component, which activates a serverless function that takes actions to reduce the threat. OpenWhisk will invoke a Kubernetes API extension and let the security enforcement operator (SEO) handle the operation [35].

### Networks

Serverless cloud computing has been employed in different networking domains [175, 188, 251, 252]. In [188], a variety of networking fields including software-defined networking (SDN) which can utilize advantages of serverless computing architecture have been discussed. The

SDN is a network architecture approach that enables the network to be manageable and adaptive. This architecture separates the network control plane from the forwarding functions (the data plane). This decoupling enables network switches to become a simple forwarding device, and the network control is implemented as a network application that executed on a logically centralized controller. Serverless computing can be used in the SDN controllers. These controllers can be implemented as independent functions deployed on serverless platforms. For example, when a packet arrives to the SDN forwarding device, the device will parse the packet header and forward it to the SDN controller. The functions within the SDN controller will be activated then it will determine what action to be taken with the packet. After that, it will send the information to the forwarding device. The action might be modifying the header, dropping the packet, etc.

### IoT

Serverless computing has been utilized in many IoT applications, as shown in Table 8. For example, a camera can be installed to monitor a house, after which processing images captured by the camera can be performed by some serverless functions provided by the OpenWhisk platform. When a camera detects an interesting object such as a car or a human, the camera sends its pictures to the serverless platform for further processing. To extract features, a serverless function is called to perform feature extraction and then reports its status to the users [232].

### Edge computing

Serverless cloud computing and edge computing have been used to build different kinds of applications, as presented in Table 8. For instance, the authors of [217] have implemented an autonomous mobile robot (AMR) system based on serverless computing and edge computing. The system consists of three main components: an AMR with NVIDIA Jetson TX2 module for edge computing, a serverless architecture based on AWS, and a cross-platform mobile application developed using React Native. The main idea of the system is to deliver a package to a user. For example, the user will interact with the

**Table 8** Top three serverless application areas

Application Area	Papers	Total
Edge Computing	[6, 23, 60, 62, 140, 144, 145, 148, 151, 158, 161, 162, 189, 193, 217, 222, 231, 232, 235, 237, 253–257]	25
Security	[12, 35, 94, 97, 104, 114, 124, 135, 146, 147, 176, 178, 187, 221, 256, 258–262]	20
IoT	[1, 10, 60, 77, 144, 151, 158, 162, 222, 232, 234, 237, 241, 255–257, 260, 263, 264]	19

mobile application to send a package. Once the delivery request has been received from the user, the AWS IoT can activate related lambda functions, such as position coordinate. Then, the AMR would start its mission, sending the package to the receiver's location. Also, facial images were regularly retrieved by AWS lambda to identify the receiver's face. Finally, the task is completed when the receiver's identification is confirmed [217].

#### Serverless computing challenges and issues (RQ7)

Studying the literature reveals a number of challenges and issues posed by employing serverless computing. These challenges cover the functional and non-functional aspects of serverless computing as follows:

##### Cost and pricing model

Cost is a fundamental challenge; therefore, serverless computing providers should reduce the usage of resources to the minimum, while functioning in both execution and idle states. Further, the pricing model is another challenge in serverless computing compared to other cloud computing approaches. For example, the CPU bound is cheap, whereas the input/output (I/O) bound functions may be more expensive from dedicated servers. Table 9 presents papers that investigate issues on cost and pricing models in serverless cloud computing.

##### Cold start

Serverless computing can scale to zero while there is no request for functions and services. Scaling to zero leads to a problem called cold start. A cold start occurs when serverless functions remain idle for some time, and the next time these functions are invoked, a longer start time is required. Methods and techniques to reduce the cold start problem are crucial as a result, many papers have been studied this problem, as shown in Table 9.

##### Resource limits

In serverless computing, resources are required to ensure that the platform can deal with load increasing. This

includes CPU usage, memory, execution time, and bandwidth [94, 202, 210, 235, 280].

##### Security

Security is the most challenging issue in serverless cloud computing. One of the security issues is isolation, because functions are running on a shared platform by many users. Therefore, strong isolation is required. Another security issue is trust when it comes to process-sensitive data. The serverless applications work with many system components, which must function correctly to maintain security properties. Table 9 presents several papers associated with serverless security.

##### Scalability

Serverless computing must ensure function scalability and elasticity. For example, when many requests are issued to a serverless application, these requests should all be served and the used serverless cloud provider should provide the required resources to process all these requests and should scale up with the number of requests [210, 280, 281].

##### Long-running

Serverless computing runs function in a limited and short execution time, while there are some tasks might require long execution time. This does not support long execution running, since these functions are stateless, which means that if the function is paused it cannot be resumed again [11, 202, 234, 280].

##### Programming & debugging

There is currently a lack of debugging tools. Further, monitoring tools are required, since developers need to monitor the application and observe how functions are working. More advanced integrated development environments (IDEs) are needed, so developers can perform refactoring functions, such as merging or splitting functions, and reverting functions to the previous version. Moreover, logs from serverless function invocations need to be sent to the developer and provide detailed stack traces. When an error occurs, a good method is required to report details on the occurrence to the developer. The equivalent of a stack trace for serverless computing is currently not available. Table 9 shows many papers that consider programming and debugging challenges and issues.

##### Vendor lock-in

The FaaS paradigm separates the code from the data, which leads the functions to depend strongly on the cloud provider's ecosystem for storing, obtaining, and transferring data [282]. This issue makes the customers dependent on the serverless provider for products and services, and the customers cannot easily use different vendors in the

**Table 9** Top challenges and issues

Challenges and Issues	Paper	Total
Performance	[11, 16–18, 44, 81, 84, 85, 89–91, 95, 96, 114, 126, 157, 161, 162, 203, 204, 210, 219, 226, 227, 232, 262, 265–271]	33
Cold start	[91, 98, 100, 105, 137, 149, 172, 202–204, 210, 229, 232, 272–274, 280]	17
Security	[11, 17, 44, 84, 97, 98, 114, 187, 210, 256, 258, 275, 280]	13
Programming and debugging	[9, 11, 15, 16, 44, 101, 119, 120, 210, 235, 276, 277, 280]	13
Cost and pricing	[11, 16, 81, 88, 202, 204, 210, 267, 278–280]	11

future without substantial cost. Thus, customers have to wait on the serverless provider for additional services [9, 130, 202].

#### **Performance**

Serverless computing has many performance challenges and issues such as scheduling and service calling overhead. For instance, scheduling means when a serverless function is activated in response to an event this function should be mapped to a specific resource (e.g., container or VM) to be run. The resource can have a significant effect on performance based on available resources, location of input data and code, load balancing, etc. Table 9 shows papers related to serverless performance.

#### **Fault tolerance**

It refers to a system that continues working and provides its services despite the failure in some components. It mostly occurs when some containers fail. To overcome this challenge, a basic retry mechanism is used [11, 210, 235].

#### **Function composition**

Serverless cloud vendors provide users the ability to deploy small stateless functions to the cloud to handle a specific task. However, some complex tasks require multiple functions to work with each other collaboratively to be performed. Therefore, more research needs to be done on how function composition can be used effectively and efficiently in serverless cloud computing [11, 38, 235].

#### **Resource sharing**

Functions in serverless cloud computing share resources to achieve inexpensive cloud computing. Sharing resources among functions and other serverless components is a challenging task. Therefore, good techniques are required to be investigated to achieve this goal [98, 210, 283].

#### **Testing**

A serverless application consists of many small functions. These functions work together to accomplish the application's functionality. Therefore, integration testing for these functions is a crucial issue to make sure that the application works properly [9, 84, 284].

#### **Naming and addressing system**

Users deploy functions to serverless cloud computing to solve problems. These functions cannot listen to network communications. The existing serverless cloud computing frameworks do not support this feature. Instead, they use third party services such as Amazon S3 to communicate with other functions. Therefore, finding the internet protocol (IP) address of a function by other functions and services is a challenging issue in serverless cloud computing [98].

#### **Legacy systems**

Legacy systems refer to old technologies, techniques, hardware, and software systems that are still in use. It should be possible to reach these systems from serverless cloud computing. Also, these systems might be required to be transferred to cloud computing. Therefore, more work needs to be done on the migration process and how the functions can be extracted from legacy systems to be deployed as serverless cloud functions [84, 119, 120, 210, 280].

#### **Managing hybrid cloud**

In a hybrid cloud, a developer may deploy an application to different clouds (hybrid cloud). For example, if some functions of an application are on a specific serverless cloud vendor and others are hosted on other public clouds; then, managing these functions and their interactions is a challenging issue [84, 210, 280].

#### **Lack of quality of service (QoS) support**

Existing serverless platforms and frameworks do not provide users the control over the QoS of serverless functions [235]. Cold starts, queuing, and orchestration are the main reasons affecting the QoS in serverless computing [8].

#### **Architecture complexity**

A serverless application may consist of several functions; the number of functions increases the complexity of the architecture. Managing these functions and services related to the application also leads to a complex architecture [9].

#### **Interactions tracking**

Stateful requests are usually used by real-life applications. It means deployed systems keep track of the state of users' interactions and store them on the server-side for further uses. However, in stateless serverless functions, it is not obvious how these functions will handle and manage the states of each user [210, 280].

#### **Concurrency management**

Concurrency means a function can handle any number of requests whenever a function is invoked. For example, if a request has been made to a serverless function, the function will process that request. However, if another request has been made to that function and the function is still processing the previous request, then the serverless should provide another instance of that function to serve the new request [210, 280].

#### **Support for heterogeneous hardware**

Existing serverless platforms may not support some specialized hardware such as graphics processing unit (GPU) and field programmable gate arrays (FPGAs). This is a challenging issue for vendors to provide support for heterogeneous hardware [210, 280].



### Tools available for serverless computing (RQ8)

Nowadays, various providers strive to facilitate the adjustable use and allocation of machine resources on the cloud [9]. Likewise, plenty of supportive tools and services are aiding developers to more efficiently manage and deploy applications using serverless computing. Serverless computing is auto-scalable, reliable, and easily accessible [203]; for these reasons, big cloud providers such as Amazon, Microsoft, Google, IBM have realized the importance of offering frameworks, IDEs, software development kits (SDKs), function development kits (FDK), migrating mechanisms, logs, and monitoring tools to enhance and simplify the development, testing, deployment, and monitoring of serverless applications [17]. For instance, Amazon offers Cloud9 IDE for local deploying and testing [205].

Apart from the cloud providers' specific tools, plenty of third-party tools exist for the developers. With the concept of these tools, developers can build and deploy applications on multi-cloud providers. Developers are also able to control platforms and resources by programming. The advantages of this are linking the applications with auto-scaling controllers and including advanced self-mechanisms into the code to automatically configure, secure, optimize, and recover the cloud applications. The core advantage of this feature is the acceleration in applying changes to the application environment [272].

There are several tools available to model serverless applications, which are based on deployment models as either imperative or declarative. The imperative model defines the execution steps to obtain a specific deployment task. While the declarative model describes the structure of a desired application deployment. However, to fully benefit from employing a serverless architecture, cloud providers should address issues that have arisen with the use of a serverless paradigm. For instance, debugging tools are unable to track and identify the exact reason behind errors [44], as most of them are limited to what cloud providers offer [179]. Although many powerful tools have been mentioned in this study and can be used in serverless computing in real scenarios, there is still a great opportunity to develop further tools and services.

### Migration of monolithic applications to serverless computing (RQ9)

The nature of most existing applications is monolithic. Monolithic applications have several drawbacks; they are characterized by continuous growth in complexity and size over time.

The bigger size of the monolithic applications leads to slower startup time. Moreover, novice developers face difficulties in digesting the traditional programming paradigm. Economically, monolithic systems take more effort to be developed and debugged. Furthermore,

integrating the latest technological development into monolithic systems is a tough and expensive process. Generally, monolithic applications are designed to be tightly coupled – the entire application will be unable to run or compile if one component is missing or fails [128]. It is also difficult to scale the application when multiple components have limited resources.

Another drawback is that updating any component will require redeployment of the entire project. The migration process to serverless computing involves transferring the legacy application code to serverless functions. This process could be more efficient and functional in applications with less size [76].

The key challenging aspect of migration is about extracting the serverless computing from the monolithic systems. There are several approaches to accomplish this task, one of them is Lift and shift [205]. This technique transfers the whole infrastructure to the cloud, however, this method also brings the already existing problems within the source to the destination. In [205] the authors proposed toLambda to automatically refactor, test, and deploy the monolithic applications (Java) into microservices (AWS Lambda Node.js). While rebuilding the legacy application from scratch is recommended for applications that no longer depend on the existing cloud services [130].

However, not all applications are suitable for migration to serverless computing [76, 128]; therefore, the first important aspect to be considered before rebuilding the applications is whether it would save money [188]. For such cases, newly desired features could be implemented and added via serverless computing as an extension to the current systems [128].

The other approach is to refactor the entire legacy code into FaaS services. During the migration phase, it is crucial to address the coupling of the systems not only in the application logic but also in the databases, as more functions will call the same database. However, migrating the server-side while keeping the user interface could lead to problems. Moreover, the client cannot obtain integrated data by a single request. As the functions are decoupled into smaller entities, the server is unable to aggregate data from different entities. Thus, it is the client's responsibility to call the necessary entities to achieve this task [76].

### Future directions of research (RQ10)

As the evolution of serverless computing is relatively new, there are several research paths available to be focused on as follows:

#### Function startup

One of the major research opportunities is overcoming the cold start problem without affecting the primary feature of serverless which is scaling to zero [160, 188]. The first call of functions needs initializing



the required libraries, which will cause a cold start. To bypass this, the computing resources will be warm for a certain time. Hence, upcoming requests will be handled faster. This could be performed via enhancing scheduling policies and developing more accurate function performance measurements [86]. Serverless providers follow their approaches to keep the functions in the warm pool. However, most of them are based on the number of requests for a certain time. Thus, if a function is not called frequently, it will suffer again from the cold start.

Very few studies such as [272] suggested a periodic event scheduler for Kotless (a serverless framework for Kotlin) which will trigger a list of warm functions every few minutes. The authors of the study claimed that this will reduce the cold start without bringing extra costs. While in [233] argues that pre-warming methods are unnecessarily using resources with idle containers. The researchers are still working to avoid cold start by reducing high delayed function startups via optimizing compute resources [11].

Recycling and rebalancing minutes and hours of idle runtime is an expensive process for cloud providers. Therefore, reducing the cold start penalties will help cloud providers in the first place and hence customers. The authors in [202] proposed FaaSTest an autonomous approach based on machine learning to capture the function call behavior and then dynamically select the optimal ones. This technique could reduce the cold start by 90%. They proposed a strategy to predict functions invoking time and warming the function using fine-grained regression method [285]. However, overcoming the issue of function startups is still considered as a research direction to be more investigated.

### QoS

Keeping a guaranteed QoS level in the software level agreement (SLA) that describes the lower service level offered by the service providers [166] is a major obstacle for cloud providers to offer optimal performance metrics [167, 207]. However, serverless frameworks should consider the objectives of both providers and users [242]; customers and developers have none or little QoS support over the functions [236]. In addition, the auto-scaling feature lacks QoS guarantees. This lack of QoS affects the performance of serverless applications. Increasing response time leads to decreasing the QoS level [207]. It also raises the cost of the service [236]. Therefore, achieving an ideal resource allocation management is a complicated and challenging task as several objectives should be fulfilled together [209]. Hence, providing more efficient QoS management of functions by the auto-scaling is essential to be considered without degrading the fault-tolerance features and increasing the cost.

### Pricing

Pricing is crucial for both customers and cloud providers. However, there is a shortage in pricing models, as there is an imbalance in needs between serverless providers, developers, and service end-users [236]. The pricing scheme for most cloud providers is based on the number of functions' requests and execution time-the quantity of consumed resources [123, 200]. Currently, FaaS is less expensive when functions are bound to I/O than CPU. Moreover, services that dynamically adjust resource consumption are unable to predict the optimal computing technology. It is crucial to implement solutions that offer cost-effective computing resources. FaaSTest reduced the cost by 50% via learning the behavioral pattern of functions using machine learning [202]. Price estimation has a great impact on selecting the most optimal provider. Therefore there should be more researches on developing tools to predict the pricing in advance.

### Legacy systems

Since the serverless emergence, researchers are working on the open question of how to decompose legacy systems into FaaS without degrading performance [208]. Several works have been done on migrating to FaaS [76, 130, 286]. The currently available automated tools for migrating legacy code into FaaS are not fully practical due to the remaining manual work that needs to be done [17]. Therefore, finding optimal automatic migration solutions for existing legacy systems is an interesting research direction [130]. Moreover, research on tools for checking whether a legacy system will fit the serverless paradigm is a crucial line. Also, developing and enhancing automatic and semi-automatic analysis strategies based on artificial intelligence could be another future research field.

### Debugging, testing, and benchmarking

The available tools for testing, debugging, and deployment are immature, this prevents some developers from entering the serverless environment. The shortage of tools in FaaS is a core problem, particularly the testing tools [17]. Moreover, most FaaS environments lack powerful local emulation platforms for testing. Therefore, developers are mostly depending on the server-side, which is expensive. Developers need to be ensured about the adequate testing tools before diving into the serverless world. A challenging aspect in benchmarking is the lack of information due to the heterogeneity of the cloud provider data center: hardware, software, and configurations [287]. Additionally, benchmarking FaaS platforms should take advantage of analyzing the cloud services, which lacks limited accessible measurements and hidden modification of services over time [55]. Thus, it is essential to have transparent, fair, and standardized benchmarking tools available for developers.

## Threats to validity

Several threats might impact the validity of the literature mapping studies. In this paper, popular instructions and guidelines were taken into account to avoid threats to validity as follows:

- Coverage of research questions: All up-to-date research aspects of serverless computing might not be included in this study. To overcome this threat, the brainstorming was conducted by all the authors in determining the most current research questions in the area.
- Coverage of related papers: The process of obtaining all the related studies in serverless computing cannot be secured. In this study, various literature databases were employed; moreover, the method based on different terms and synonymous is followed by all the authors in determining the related questions.
- Paper inclusion/exclusion criteria: The individual bias and interpretation could affect the implementation of the criteria. Therefore, to solve this problem, the agreements of all authors were considered in excluding or including a paper.
- Accuracy of data extraction: The individual experience effects extracting the data, therefore online meetings were conducted after the data extraction process by each author. During the meetings, the outcomes from each author were compared with other findings to determine the differences and reach a final consensus.
- Reproducibility of the study: Whether other researchers could obtain similar outcomes of this study is another threat. Thus, to address this, the research methodology contains the well-explained steps and actions conducted in this paper (as shown in “[Research methodology](#)” section).

## Conclusions

The contributions of the work presented in this paper are threefold: (a) a methodical review of related literature on the topic of serverless computing, to address the issue of the lack of compiling information on the state-of-the-art of the field; (b) a comparison of the platforms and tools used in serverless computing; (c) an extensive analysis of the differences, benefits, and issues related to serverless computing, to provide a more complete understanding of the topic. Given the fast evolution and growing interest in the field, this survey focused on gathering the most outstanding trends and outcomes of serverless computing, as described by recent researchers. This survey could significantly reduce ambiguity and the entry barrier for novice developers to adapt to the serverless environment. Furthermore, the findings presented in this study could be of great value for future researchers interested

in further investigating serverless computing. Finally, it is worth mentioning that the interest that both commercial and academic efforts fueled into studying, developing, and implementing serverless tools in forthcoming years could help maximize the potential that serverless computing could bring to the IT community.

## Acknowledgements

Not applicable.

## Authors' contributions

Conceptualization: HBH, SAB, and QIS; methodology: HBH, SAB, and QIS; validation: HBH, SAB, and QIS; formal analysis: HBH, SAB, and QIS; investigation: HBH, SAB, and QIS; resources: HBH; data curation, HBH and SAB; writing—original draft preparation: HBH, SAB, and QIS; writing—review and editing: HBH, SAB, and QIS; visualization: SAB; supervision: QIS; It is noted that all authors cooperated with each other to achieve suitable information flow across the entire paper. The authors read and approved the final manuscript.

## Authors' information

**Hassan B. Hassan** received the B.Sc. degree in Computer Science from University of Duhok, Iraq, in 2010. He completed the M.Sc. degree in Web Applications and Services, from Leicester University, UK, in 2015. He is currently working as an assistant lecturer at the college of medicine, University of Duhok, Iraq. His main areas of research interest are cloud computing, web programming, big data, and human computer interaction.

**Saman A. Barakat** received the B.Sc. degree in Computer Science from University of Duhok, Iraq, in 2008. He completed the M.Sc. degree in Advanced Computer Science, from Newcastle University, UK, in 2012. He is currently working as a lecturer at the college of science, University of Duhok, Iraq. His main areas of research interest are cloud computing, and software engineering.

**Qusay I. Sarhan** received the B.Sc. degree in Software Engineering from University of Mosul, Iraq, in 2007 and the M.Tech. degree in Software Engineering from Jawaharlal Nehru Technological University, India, in 2011. Currently, he is a lecturer and the leader of Software Engineering and Embedded Systems (SEES) research group at University of Duhok, Iraq. He has a couple of national and international publications and his research interests include software engineering, internet of things, and embedded systems.

## Funding

Not applicable.

## Availability of data and materials

Not applicable.

## Declarations

### Competing interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>Software Engineering and Embedded Systems (SEES) Research Group, College of Medicine, University of Duhok, Duhok, Kurdistan Region, Iraq.

<sup>2</sup>Software Engineering and Embedded Systems (SEES) Research Group, Department of Computer Science, College of Science, University of Duhok, Duhok, Kurdistan Region, Iraq.

Received: 1 July 2020 Accepted: 21 June 2021

Published online: 12 July 2021

## References

1. Großmann M, Ioannidis C, Le DT (2019) Applicability of serverless computing in fog computing environments for iot scenarios. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC '19 Companion). Association for Computing Machinery, New York. pp 29–34. <https://doi.org/10.1145/3368235.3368834>

2. Boza EF, Abad CL, Villavicencio M, Quimba S, Plaza JA (2017) Reserved, on demand or serverless: Model-based simulations for cloud budget planning. In: 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). pp 1–6. <https://doi.org/10.1109/ETCM.2017.8247460>
3. Villamizar M, Garcés O, Ochoa L, Castro H, Salamanca L, Verano M, Casallas R, Gil S, Valencia C, Zambrano A, Lang M (2017) Cost comparison of running web applications in the cloud using monolithic, microservice, and aws lambda architectures. SOCA 11(2):233–247. <https://doi.org/10.1007/s11761-017-0208-y>
4. Jonas E, Schleier-Smith J, Sreekanti V, Tsai C-C, Khandelwal A, Pu Q, Shankar V, Carreira J, Krauth K, Yadwadkar N, Gonzalez JE, Popa RA, Stoica I, Patterson DA (2019) Cloud Programming Simplified: A Berkeley View on Serverless Computing. <http://arxiv.org/abs/1902.03383>. Accessed 6 Jan 2021
5. Geng X, Ma Q, Pei Y, Xu Z, Zeng W, Zou J (2018) Research on early warning system of power network overloading under serverless architecture. In: 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2). pp 1–6. <https://doi.org/10.1109/EI2.2018.8582355>
6. Kulkarni SG, Liu G, Ramakrishnan KK, Wood T (2019) Living on the edge: Serverless computing and the cost of failure resiliency. In: 2019 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN). pp 1–6. <https://doi.org/10.1109/LANMAN.2019.8846970>
7. Baldini I, Castro P, Chang K, Cheng P, Fink S, Ishakian V, Mitchell N, Muthusamy V, Rabbah R, Slominski A, Suter P (2017) Serverless Computing: Current Trends and Open Problems. In: Research Advances in Cloud Computing. Springer, Singapore. pp 1–20. [https://doi.org/10.1007/978-981-10-5026-8\\_1](https://doi.org/10.1007/978-981-10-5026-8_1)
8. Adzic G, Chatley R (2017) Serverless computing: Economic and architectural impact. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017). Association for Computing Machinery, New York. pp 884–889. <https://doi.org/10.1145/3106237.3117767>
9. Jambunathan B, Yoganathan K (2018) Architecture decision on using microservices or serverless functions with containers. In: 2018 International Conference on Current Trends Towards Converging Technologies (ICCTCT). pp 1–7. <https://doi.org/10.1109/ICCTCT.2018.8551035>
10. Wolski R, Krintz C, Bakir F, George G, Lin W-T (2019) Cspot: Portable, multi-scale functions-as-a-service for iot. In: Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (SEC '19). Association for Computing Machinery, New York. pp 236–249. <https://doi.org/10.1145/3318216.3363314>
11. Yussupov V, Breitenbücher U, Leymann F, Wurster M (2019) A systematic mapping study on engineering function-as-a-service platforms and tools. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'19). Association for Computing Machinery, New York. pp 229–240. <https://doi.org/10.1145/3344341.3368803>
12. Brenner S, Kapitza R (2019) Trust more, serverless. In: Proceedings of the 12th ACM International Conference on Systems and Storage (SYSTOR '19). Association for Computing Machinery, New York. pp 33–43. <https://doi.org/10.1145/3319647.3325825>
13. Kuhlenskamp J, Werner S (2018) Benchmarking faas platforms: Call for community participation. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). pp 189–194. <https://doi.org/10.1109/UCC-Companion.2018.00055>
14. Somma G, Ayimba C, Casari P, Romano SP, Mancuso V (2020) When less is more: Core-restricted container provisioning for serverless computing. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp 1153–1159. <https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162876>
15. Sewak M, Singh S (2018) Winning in the era of serverless computing and function as a service. In: 2018 3rd International Conference for Convergence in Technology (I2CT). pp 1–5. <https://doi.org/10.1109/I2CT.2018.8529465>
16. van Eyk E, Toader L, Talluri S, Versluis L, Ut A, Iosup A (2018) Serverless is more: From paas to present cloud computing. IEEE Internet Comput 22(5):8–17. <https://doi.org/10.1109/MIC.2018.053681358>
17. Leitner P, Wittern E, Spillner J, Hummer W (2019) A mixed-method empirical study of function-as-a-service software development in industrial practice. J Syst Softw 149:340–359. <https://doi.org/10.1016/j.jss.2018.12.013>
18. Feng L, Kudva P, Da Silva D, Hu J (2018) Exploring serverless computing for neural network training. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). pp 334–341. <https://doi.org/10.1109/CLOUD.2018.00049>
19. Ao L, Izhikevich L, Voelker GM, Porter G (2018) Sprocket: A serverless video processing framework. In: Proceedings of the ACM Symposium on Cloud Computing (SoCC '18). Association for Computing Machinery, New York. pp 263–274. <https://doi.org/10.1145/3267809.3267815>
20. Werner S, Kuhlenskamp J, Klems M, Müller J, Tai S (2018) Serverless big data processing using matrix multiplication as example. In: 2018 IEEE International Conference on Big Data (Big Data). pp 358–365. <https://doi.org/10.1109/BigData.2018.8622362>
21. Al-Ali Z, Goodarzi S, Hunter E, Ha S, Han R, Keller E, Rozner E (2018) Making serverless computing more serverless. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). pp 456–459. <https://doi.org/10.1109/CLOUD.2018.00064>
22. Pérez A, Risco S, Naranjo DM, Caballer M, Moltó G (2019) On-premises serverless computing for event-driven data processing applications. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). pp 414–421. <https://doi.org/10.1109/CLOUD.2019.00073>
23. Glikson A, Nastic S, Dustdar S (2017) Deviceless edge computing: Extending serverless computing to the edge of the network. In: Proceedings of the 10th ACM International Systems and Storage Conference (SYSTOR '17). Association for Computing Machinery, New York. <https://doi.org/10.1145/3078468.3078497>
24. Al-Ameen M, Spillner J (2019) A systematic and open exploration of faas research. In: Proceedings of the European Symposium on Serverless Computing and Applications (CEUR Workshop Proceedings ; 2330). CEUR-WS, Zurich. pp 30–35. <https://doi.org/10.21256/zhaw-3271>
25. Alqaryouti O, Siyam N (2018) Serverless computing and scheduling tasks on cloud: A review. Am Sci Res J Eng Technol Sci (ASRJETS) 40(1):235–247
26. Taibi D, El Ioini N, Pahl C, Niederkofer J (2020) Patterns for Serverless Functions (Function-as-a-Service): A Multivocal Literature Review. In: Proceedings of the 10th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER. pp 181–192. <https://doi.org/10.5220/0009578501810192>
27. Hellerstein JM, Faleiro J, Gonzalez JE, Schleier-Smith J, Sreekanti V, Tumanov O, Wu C (2018) Serverless Computing: One Step Forward, Two Steps Back. <http://arxiv.org/abs/1812.03651>. Accessed 4 Oct 2021
28. Rajan AP (2020) A review on serverless architectures - function as a service (faas) in cloud computing. Telecommun Comput Electron Control 18(1):530–537. <https://doi.org/10.12928/telkomnika.v18i1.12169>
29. Sadaqat M, Colomo-Palacios R, Knudsen LES (2018) Serverless Computing: A Multivocal Literature Review. NOKOBIT - Norsk Konferanse for Organisasjoners Bruk Av Informasjonsteknologi 26(1):1–13
30. Scheuner J, Leitner P (2020) Function-as-a-service performance evaluation: A multivocal literature review. J Syst Softw 170:110708. <https://doi.org/10.1016/j.jss.2020.110708>
31. Fox GC, Ishakian V, Muthusamy V, Slominski A (2017) Status of Serverless Computing and Function-as-a-Service(FaaS) in Industry and Research. arXiv e-prints:1708–08028. <http://arxiv.org/abs/1708.08028>. Accessed 6 Jan 2021
32. Hassan HB, Barakat SA, Sarhan QI (2021) Serverless Literature Dataset. Zenodo. <https://doi.org/10.5281/zenodo.4660553>
33. Pedreira O, García F, Brisaboa N, Piattini M (2015) Gamification in software engineering - a systematic mapping. Inf Softw Technol 57:157–168
34. Lopez-herrejon RE, Linsbauer L, Egyed A (2015) A systematic mapping study of search-based software engineering for software product lines. Inf Softw Technol 61:33–51
35. Bila N, Dettori P, Kansa A, Watanabe Y, Youssef A (2017) Leveraging the serverless architecture for securing linux containers. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW). pp 401–404. <https://doi.org/10.1109/ICDCSW.2017.66>
36. Chang KS, Fink SJ (2017) Visualizing serverless cloud application logs for program understanding. In: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). pp 261–265. <https://doi.org/10.1109/VLHCC.2017.8103476>
37. Ishakian V, Muthusamy V, Slominski A (2018) Serving deep learning models in a serverless platform. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). pp 257–262. <https://doi.org/10.1109/IC2E.2018.00052>

38. Baldini I, Cheng P, Fink SJ, Mitchell N, Muthusamy V, Rabbah R, Suter P, Tardieu O (2017) The serverless trilemma: Function composition for serverless computing. In: Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! 2017). Association for Computing Machinery, New York. pp 89–103. <https://doi.org/10.1145/3133850.3133855>
39. Kanso A, Youssef A (2017) Serverless: Beyond the cloud. In: Proceedings of the 2nd International Workshop on Serverless Computing (WoSC '17). Association for Computing Machinery, New York. pp 6–10. <https://doi.org/10.1145/3154847.3154854>
40. Koller R, Williams D (2017) Will serverless end the dominance of linux in the cloud? In: Proceedings of the 16th Workshop on Hot Topics in Operating Systems (HotOS '17). Association for Computing Machinery, New York. pp 169–173. <https://doi.org/10.1145/3102980.3103008>
41. Mukhi NK, Prabhu S, Slawson B (2017) Using a serverless framework for implementing a cognitive tutor: Experiences and issues. In: Proceedings of the 2nd International Workshop on Serverless Computing (WoSC '17). Association for Computing Machinery, New York. pp 11–15. <https://doi.org/10.1145/3154847.3154852>
42. Nadgowda S, Bila N, Isci C (2017) The less server architecture for cloud functions. In: Proceedings of the 2nd International Workshop on Serverless Computing (WoSC '17). Association for Computing Machinery, New York. pp 22–27. <https://doi.org/10.1145/3154847.3154850>
43. Klimovic A, Wang Y, Stuedi P, Trivedi A, Pfefferle J, Kozyrakis C (2018) Pocket: Elastic ephemeral storage for serverless analytics. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'18). USENIX Association, USA. pp 427–444
44. Yan M, Castro P, Cheng P, Ishakian V (2016) Building a chatbot with serverless computing. In: Proceedings of the 1st International Workshop on Mashups of Things and APIs (MOTA '16). Association for Computing Machinery, New York. pp 1–4. <https://doi.org/10.1145/3007203.3007217>
45. Barcelona-Pons D, García-López P, Ruiz A, Gómez-Gómez A, París G, Sánchez-Artigas M (2019) Faas orchestration of parallel workloads. In: Proceedings of the 5th International Workshop on Serverless Computing (WoSC '19). Association for Computing Machinery, New York. pp 25–30. <https://doi.org/10.1145/3366623.3368137>
46. Barcelona-Pons D, Sánchez-Artigas M, París G, Sutra P, García-López P (2019) On the faas track: Building stateful distributed applications with serverless architectures. In: Proceedings of the 20th International Middleware Conference (Middleware '19). Association for Computing Machinery, New York. pp 41–54. <https://doi.org/10.1145/3361525.3361535>
47. Kaviani N, Kalinin D, Maximilien M (2019) Towards serverless as commodity: A case of knative. In: Proceedings of the 5th International Workshop on Serverless Computing (WoSC '19). Association for Computing Machinery, New York. pp 13–18. <https://doi.org/10.1145/3366623.3368135>
48. Byrne A, Nadgowda S, Coskun AK (2020) Ace: Just-in-time serverless software component discovery through approximate concrete execution. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 37–42. <https://doi.org/10.1145/3429880.3430098>
49. Sánchez-Artigas M, Eizaguirre GT, Vernik G, Stuart L, García-López P (2020) Primula: A Practical Shuffle/Sort Operator for Serverless Computing. Association for Computing Machinery, New York
50. Parás G, García-López P, Sánchez-Artigas M (2020) Serverless elastic exploration of unbalanced algorithms. In: 2020 IEEE 13th International Conference on Cloud Computing (CLOUD). pp 149–157. <https://doi.org/10.1109/CLOUD49709.2020.00033>
51. López PG, Arjona A, Sampé J, Slominski A, Villard L (2020) Triggerflow: Trigger-based orchestration of serverless workflows. In: Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems (DEBS '20). Association for Computing Machinery, New York. pp 3–14. <https://doi.org/10.1145/3401025.3401731>
52. Carver B, Zhang J, Wang A, Anwar A, Wu P, Cheng Y (2020) Wukong: A scalable and locality-enhanced framework for serverless parallel computing. In: Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20). Association for Computing Machinery, New York. pp 1–15. <https://doi.org/10.1145/3419111.3421286>
53. Klimovic A, Wang Y, Kozyrakis C, Stuedi P, Pfefferle J, Trivedi A (2018) Understanding ephemeral storage for serverless analytics. In: Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '18). USENIX Association, USA. pp 789–794
54. Wang A, Zhang J, Ma X, Anwar A, Rupprecht L, Skourtis D, Tarasov V, Yan F, Cheng Y (2020) Infinicache: Exploiting ephemeral serverless functions to build a cost-effective memory cache. In: 18th USENIX Conference on File and Storage Technologies (FAST '20). USENIX Association, Santa Clara. pp 267–281
55. Kuhlenskamp J, Werner S, Borges MC, El Tal K, Tai S (2019) An evaluation of faas platforms as a foundation for serverless big data processing. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'19). Association for Computing Machinery, New York. pp 1–9. <https://doi.org/10.1145/3344341.3368796>
56. Werner S, Girke R, Kuhlenskamp J (2020) An evaluation of serverless data processing frameworks. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 19–24. <https://doi.org/10.1145/3429880.3430095>
57. Kuhlenskamp J, Werner S, Borges MC, Ernst D, Wenzel D (2020) Benchmarking elasticity of faas platforms as a foundation for objective-driven design of serverless applications. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20). Association for Computing Machinery, New York. pp 1576–1585. <https://doi.org/10.1145/3341105.3373948>
58. Werner S, Kuhlenskamp J, Pallas F, Anders N, Mucaj N, Tsaplina O, Schmidt K, Yildirim K (2020) Diminuendo! tactics in support of faas migrations. In: Paasivaara M, Kruchten P (eds). Agile Processes in Software Engineering and Extreme Programming – Workshops. Springer, Cham. pp 125–132
59. Kuhlenskamp J, Werner S, Tai S (2020) The ifs and buts of less is more: A serverless computing reality check. In: 2020 IEEE International Conference on Cloud Engineering (IC2E). pp 154–161. <https://doi.org/10.1109/IC2E48712.2020.00023>
60. Pfandzelter T, Bermbach D (2020) tinyfaas: A lightweight faas platform for edge environments. In: 2020 IEEE International Conference on Fog Computing (ICFC). pp 17–24. <https://doi.org/10.1109/ICFC49376.2020.00011>
61. Bermbach D, Karakaya A-S, Buchholz S (2020) Using application knowledge to reduce cold starts in faas services. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20). Association for Computing Machinery, New York. pp 134–143. <https://doi.org/10.1145/3341105.3373909>
62. Bermbach D, Maghsudi S, Hasenburger J, Pfandzelter T (2020) Towards auction-based function placement in serverless fog platforms. In: 2020 IEEE International Conference on Fog Computing (ICFC). pp 25–31. <https://doi.org/10.1109/ICFC49376.2020.00012>
63. García López P, Sánchez-Artigas M, París G, Barcelona Pons D, Ruiz Ollobarren A, Arroyo Pinto D (2018) Comparison of faas orchestration systems. <https://doi.org/10.1109/ucc-companion.2018.00049>
64. Sampé J, Sánchez-Artigas M, García-López P, París G (2017) Data-driven serverless functions for object storage. In: Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference (Middleware '17). Association for Computing Machinery, New York. pp 121–133. <https://doi.org/10.1145/3135974.3135980>
65. Sampé J, Vernik G, Sánchez-Artigas M, García-López P (2018) Serverless data analytics in the ibm cloud. In: Proceedings of the 19th International Middleware Conference Industry (Middleware '18). Association for Computing Machinery, New York. pp 1–8. <https://doi.org/10.1145/3284028.3284029>
66. Mirabelli ME, García-López P, Vernik G (2020) Bringing scaling transparency to proteomics applications with serverless computing. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 55–60. <https://doi.org/10.1145/3429880.3430101>
67. Carreira J, Fonseca P, Tumanov A, Zhang A, Katz R (2019) Cirrus: A serverless framework for end-to-end ml workflows. In: Proceedings of the ACM Symposium on Cloud Computing (SoCC '19). Association for Computing Machinery, New York. pp 13–24. <https://doi.org/10.1145/3357223.3362711>
68. Sreekanti V, Wu C, Chhatrapati S, Gonzalez JE, Hellerstein JM, Faleiro JM (2020) A fault-tolerance shim for serverless computing. In: Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys



- '20). Association for Computing Machinery, New York. <https://doi.org/10.1145/3342195.3387535>
69. Ichnowski J, Lee W, Murta V, Paradis S, Alterovitz R, Gonzalez JE, Stoica I, Goldberg K (2020) Fog robotics algorithms for distributed motion planning using lambda serverless computing. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp 4232–4238. <https://doi.org/10.1109/ICRA40945.2020.9196651>
70. Zhang W, Fang V, Panda A, Shenker S (2020) Kappa: A programming framework for serverless computing. In: Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20). Association for Computing Machinery, New York. pp 328–343. <https://doi.org/10.1145/3419111.3421277>
71. Shankar V, Krauth K, Vodrahalli K, Pu Q, Recht B, Stoica I, Ragan-Kelley J, Jonas E, Venkataraman S (2020) Serverless linear algebra. In: Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20). Association for Computing Machinery, New York. pp 281–295. <https://doi.org/10.1145/3419111.3421287>
72. Gupta V, Carrano D, Yang Y, Shankar V, Courtade T, Ramchandran K (2020) Serverless straggler mitigation using error-correcting codes. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). pp 135–145. <https://doi.org/10.1109/ICDCS47774.2020.00019>
73. Wu C, Sreekanti V, Hellerstein JM (2020) Transactional causal consistency for serverless computing. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20). Association for Computing Machinery, New York. pp 83–97. <https://doi.org/10.1145/3318464.3389710>
74. Pu Q, Venkataraman S, Stoica I (2019) Shuffling, fast and slow: Scalable analytics on serverless infrastructure. In: Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation (NSDI'19). USENIX Association, USA. pp 193–206
75. Lloyd W, Ramesh S, Chinthalapati S, Ly L, Pallickara S (2018) Serverless computing: An investigation of factors influencing microservice performance. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). pp 159–169. <https://doi.org/10.1109/IC2E.2018.00039>
76. Lloyd W, Vu M, Zhang B, David O, Leavesley G (2018) Improving application migration to serverless computing platforms: Latency mitigation with keep-alive workloads. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). pp 195–200. <https://doi.org/10.1109/UCC-Companion.2018.00056>
77. Al-Masri E, Diabate I, Jain R, Lam MHL, Nathala SR (2018) A serverless iot architecture for smart waste management systems. In: 2018 IEEE International Conference on Industrial Internet (ICII). pp 179–180. <https://doi.org/10.1109/ICII.2018.00034>
78. Fotouhi M, Chen D, Lloyd WJ (2019) Function-as-a-service application service composition: Implications for a natural language processing application. In: Proceedings of the 5th International Workshop on Serverless Computing (WOSC '19). Association for Computing Machinery, New York. pp 49–54. <https://doi.org/10.1145/3366623.3368141>
79. Niu X, Kumanov D, Hung L-H, Lloyd W, Yeung KY (2019) Leveraging serverless computing to improve performance for sequence comparison. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '19). Association for Computing Machinery, New York. pp 683–687. <https://doi.org/10.1145/3307339.3343465>
80. Cordingly R, Yu H, Hoang V, Perez D, Foster D, Sadeghi Z, Hatchett R, Lloyd WJ (2020) Implications of programming language selection for serverless data processing pipelines. In: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PICom/CBDCCom/CyberSciTech). pp 704–711. <https://doi.org/10.1109/DASC-PICom-CBDCCom-CyberSciTech49142.2020.00120>
81. Cordingly R, Shu W, Lloyd WJ (2020) Predicting performance and cost of serverless computing functions with saaf. In: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PICom/CBDCCom/CyberSciTech). pp 640–649. <https://doi.org/10.1109/DASC-PICom-CBDCCom-CyberSciTech49142.2020.00111>
82. Cordingly R, Yu H, Hoang V, Sadeghi Z, Foster D, Perez D, Hatchett R, Lloyd W (2020) The serverless application analytics framework: Enabling design trade-off evaluation for serverless software. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 67–72. <https://doi.org/10.1145/3429880.3430103>
83. Toader L, Uta A, Musaafir A, Iosup A (2019) Graphless: Toward serverless graph processing. In: 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC). pp 66–73. <https://doi.org/10.1109/ISPDC.2019.00012>
84. van Eyk E, Iosup A, Seif S, Thömmes M (2017) The spec cloud group's research vision on faas and serverless architectures. In: Proceedings of the 2nd International Workshop on Serverless Computing (WoSC '17). Association for Computing Machinery, New York. pp 1–4. <https://doi.org/10.1145/3154847.3154848>
85. van Eyk E, Iosup A, Abad CL, Grohmann J, Eismann S (2018) A spec rg cloud group's vision on the performance challenges of faas cloud architectures. In: Companion of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18). Association for Computing Machinery, New York. pp 21–24. <https://doi.org/10.1145/3185768.3186308>
86. van Eyk E, Grohmann J, Eismann S, Bauer A, Versluis L, Toader L, Schmitt N, Herbst N, Abad CL, Iosup A (2019) The spec-rg reference architecture for faas: From microservices and containers to serverless platforms. IEEE Internet Comput 23(6):7–18. <https://doi.org/10.1109/MIC.2019.2952061>
87. van Eyk E, Scheuner J, Eismann S, Abad CL, Iosup A (2020) Beyond microbenchmarks: The spec-rg vision for a comprehensive serverless benchmark. In: Companion of the ACM/SPEC International Conference on Performance Engineering (ICPE '20). Association for Computing Machinery, New York. pp 26–31. <https://doi.org/10.1145/3375555.3384381>
88. Eismann S, Grohmann J, van Eyk E, Herbst N, Kounev S (2020) Predicting the costs of serverless workflows. In: Proceedings of the ACM/SPEC International Conference on Performance Engineering (ICPE '20). Association for Computing Machinery, New York. pp 265–276. <https://doi.org/10.1145/3358960.3379133>
89. van Eyk E, Iosup A (2018) Addressing performance challenges in serverless computing. In: ICT. OPEN
90. Hendrickson S, Sturdevant S, Harter T, Venkataramani V, Arpacı-Dusseau AC, Arpacı-Dusseau RH (2016) Serverless computation with openlambda. In: 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16). USENIX Association, Denver. pp 1–7
91. Oakes E, Yang L, Zhou D, Houck K, Harter T, Arpacı-Dusseau AC, Arpacı-Dusseau RH (2018) Sock: Rapid task provisioning with serverless-optimized containers. In: Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '18). USENIX Association, USA. pp 57–69
92. Oakes E, Yang L, Houck K, Harter T, Arpacı-Dusseau AC, Arpacı-Dusseau RH (2017) Pipsqueak: Lean lambdas with large libraries. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW). pp 395–400. <https://doi.org/10.1109/ICDCSW.2017.32>
93. Singhvi A, Khalid J, Akella A, Banerjee S (2020) Snf: Serverless network functions. In: Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20). Association for Computing Machinery, New York. pp 296–310. <https://doi.org/10.1145/3419111.3421295>
94. Wang L, Li M, Zhang Y, Ristenpart T, Swift M (2018) Peeking behind the curtains of serverless platforms. In: Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '18). USENIX Association, USA. pp 133–145
95. Abad CL, Boza EF, van Eyk E (2018) Package-aware scheduling of faas functions. In: Companion of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18). Association for Computing Machinery, New York. pp 101–106. <https://doi.org/10.1145/3185768.3186294>
96. Aumala G, Boza E, Ortiz-Avilés L, Totoy G, Abad C (2019) Beyond load balancing: Package-aware scheduling for serverless platforms. In: 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). pp 282–291. <https://doi.org/10.1109/CCGRID.2019.00042>
97. Alpernas K, Flanagan C, Fouladi S, Ryzhyk L, Sagiv M, Schmitz T, Winstein K (2018) Secure serverless computing using dynamic information flow



- control. *Proc ACM Program Lang* 2(OOPSLA):1–26. <https://doi.org/10.1145/3276488>
98. Kaffes K, Yadwadkar NJ, Kozyrakis C (2019) Centralized core-granular scheduling for serverless functions. In: *Proceedings of the ACM Symposium on Cloud Computing (SoCC '19)*. Association for Computing Machinery, New York. pp 158–164. <https://doi.org/10.1145/3357223.3362709>
99. Choi S, Shahbaz M, Prabhakar B, Rosenblum M (2019)  $\lambda$ -nic: Interactive serverless compute on smartnics. In: *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos (SIGCOMM Posters and Demos '19)*. Association for Computing Machinery, New York. pp 151–152. <https://doi.org/10.1145/3342280.3342341>
100. Manner J, Endreß M, Heckel T, Wirtz G (2018) Cold start influencing factors in function as a service. In: *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. pp 181–188. <https://doi.org/10.1109/UCC-Companion.2018.00054>
101. Manner J, Kolb S, Wirtz G (2019) Troubleshooting serverless functions: a combined monitoring and debugging approach. *SICS Softw-Intensiv Cyber-Physical Syst* 34(2):99–104. <https://doi.org/10.1007/s00450-019-00398-6>
102. Winzinger S, Wirtz G (2019) Model-based analysis of serverless applications. In: *2019 IEEE/ACM 11th International Workshop on Modelling in Software Engineering (MiSE)*. pp 82–88. <https://doi.org/10.1109/MiSE.2019.00020>
103. Winzinger S, Wirtz G (2020) Applicability of coverage criteria for serverless applications. In: *2020 IEEE International Conference on Service Oriented Systems Engineering (SOSE)*. pp 49–56. <https://doi.org/10.1109/SOSE49046.2020.00013>
104. Prechtl M, Lichtenthäler R, Wirtz G (2020) Investigating possibilities for protecting and hardening installable faas platforms. In: *Dustdar S (ed). Service-Oriented Computing*. Springer, Cham. pp 107–126
105. Gias AU, Casale G (2020) Cocoa: Cold start aware capacity planning for function-as-a-service platforms. In: *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. pp 1–8. <https://doi.org/10.1109/MASCOTS50786.2020.9285966>
106. Chatley R, Allerton T (2020) Nimbus: Improving the developer experience for serverless applications. In: *2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. pp 85–88
107. Casale G, Artač M, van den Heuvel W-J, van Hoorn A, Jakovits P, Leymann F, Long M, Papanikolaou V, Presenza D, Russo A, Srirama SN, Tamburri DA, Wurster M, Zhu L (2020) Radon: rational decomposition and orchestration for serverless computing. *SICS Softw-Intens Cyber-Phys Syst* 35(1):77–87. <https://doi.org/10.1007/s00450-019-00413-w>
108. Vandebron J, Coutinho JGF, Luk W, Nurvitadhi E, Naik M (2020) Slate: Managing heterogeneous cloud functions. In: *2020 IEEE 31st International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*. pp 141–148. <https://doi.org/10.1109/ASAP49362.2020.00032>
109. Kim J, Lee K (2019) Functionbench: A suite of workloads for serverless cloud function service. In: *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. pp 502–504. <https://doi.org/10.1109/CLOUD.2019.00091>
110. Kim J, Park J, Lee K (2019) Network resource isolation in serverless cloud function service. In: *2019 IEEE 4th International Workshops on Foundations and Applications of Self\* Systems (FAS\*W)*. pp 182–187. <https://doi.org/10.1109/FAS-W.2019.00051>
111. Park J, Kim H, Lee K (2020) Evaluating concurrent executions of multiple function-as-a-service runtimes with microvm. In: *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. pp 532–536. <https://doi.org/10.1109/CLOUD49709.2020.00080>
112. Choi J, Lee K (2020) Evaluation of network file system as a shared data storage in serverless computing. In: *Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20)*. Association for Computing Machinery, New York. pp 25–30. <https://doi.org/10.1145/3429880.3430096>
113. Kim J, Lee K (2020) I/o resource isolation of public cloud serverless function runtimes for data-intensive applications. *Clust Comput* 23(3):2249–2259. <https://doi.org/10.1007/s10586-020-03103-4>
114. Wu M, Mi Z, Xia Y (2020) A survey on serverless computing and its implications for jointcloud computing. In: *2020 IEEE International Conference on Joint Cloud Computing*. pp 94–101. <https://doi.org/10.1109/JCC49151.2020.00023>
115. Li Z, Chen Q, Xue S, Ma T, Yang Y, Song Z, Guo M (2020) Amoeba: Qos-awareness and reduced resource usage of microservices with serverless computing. In: *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. pp 399–408. <https://doi.org/10.1109/IPDPS47924.2020.00049>
116. Du D, Yu T, Xia Y, Zang B, Yan G, Qin C, Wu Q, Chen H (2020) Catalyzer: Sub-millisecond startup for serverless computing with initialization-less booting. In: *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20)*. Association for Computing Machinery, New York. pp 467–481. <https://doi.org/10.1145/3373376.3378512>
117. Yu T, Liu Q, Du D, Xia Y, Zang B, Lu Z, Yang P, Qin C, Chen H (2020) Characterizing serverless platforms with serverlessbench. In: *Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20)*. Association for Computing Machinery, New York. pp 30–44. <https://doi.org/10.1145/3419111.3421280>
118. Liu J, Mi Z, Huang Z, Hua Z, Xia Y (2020) Hcloud: A serverless platform for jointcloud computing. In: *2020 IEEE International Conference on Joint Cloud Computing*. pp 86–93. <https://doi.org/10.1109/JCC49151.2020.00022>
119. Pérez A, Moltó G, Caballer M, Calatrava A (2019) A programming model and middleware for high throughput serverless computing applications. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*. Association for Computing Machinery, New York. pp 106–113. <https://doi.org/10.1145/3297280.3297292>
120. Pérez A, Moltó G, Caballer M, Calatrava A (2018) Serverless computing for container-based architectures. *Future Generation Computer Systems* 83:50–59. <https://doi.org/10.1016/j.future.2018.01.022>
121. Giménez-Alventosa V, Moltó G, Caballer M (2019) A framework and a performance assessment for serverless mapreduce on aws lambda. *Future Generation Computer Systems* 97:259–274. <https://doi.org/10.1016/j.future.2019.02.057>
122. Naranjo DM, Risco S, de Alfonso C, Pérez A, Blanquer I, Moltó G (2020) Accelerated serverless computing based on gpu virtualization. *J Parallel Distrib Comput* 139:32–42. <https://doi.org/10.1016/j.jpdc.2020.01.004>
123. Wang H, Niu D, Li B (2019) Distributed machine learning with a serverless architecture. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. pp 1288–1296. <https://doi.org/10.1109/INFOCOM.2019.8737391>
124. Ghaemi S, Khazaei H, Musilek P (2020) Chainfaas: An open blockchain-based serverless platform. *IEEE Access* 8:131760–131778. <https://doi.org/10.1109/ACCESS.2020.3010119>
125. Goli A, Hajihassani O, Khazaei H, Ardakanian O, Rashidi M, Dauphinee T (2020) Migrating from monolithic to serverless: A fintech case study. In: *Companion of the ACM/SPEC International Conference on Performance Engineering (ICPE '20)*. Association for Computing Machinery, New York. pp 20–25. <https://doi.org/10.1145/3375555.3384380>
126. Mahmoudi N, Khazaei H (2020) Performance modeling of serverless computing platforms. *IEEE Trans Cloud Comput*:1–1. <https://doi.org/10.1109/TCC.2020.3033373>
127. Mahmoudi N, Khazaei H (2020) Temporal performance modelling of serverless computing platforms. In: *Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20)*. Association for Computing Machinery, New York. pp 1–6. <https://doi.org/10.1145/3429880.3430092>
128. Wurster M, Breitenbücher U, Képes K, Leymann F, Yussupov V (2018) Modeling and automated deployment of serverless applications using tosa. In: *2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA)*. pp 73–80. <https://doi.org/10.1109/SOCA.2018.00017>
129. Yussupov V, Breitenbücher U, Hahn M, Leymann F (2019) Serverless parachutes: Preparing chosen functionalities for exceptional workloads. In: *2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)*. pp 226–235. <https://doi.org/10.1109/EDOC.2019.00035>
130. Yussupov V, Breitenbücher U, Leymann F, Müller C (2019) Facing the unplanned migration of serverless applications: A study on portability

- problems, solutions, and dead ends. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'19). Association for Computing Machinery, New York. pp 273–283. <https://doi.org/10.1145/3344341.3368813>
131. Spillner J (2019) Serverless computing and cloud function-based applications. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC '19 Companion). Association for Computing Machinery, New York. pp 177–178. <https://doi.org/10.1145/3368235.3370269>
132. Murphy S, Persaud L, Martini W, Bosshard B (2020) On the use of web assembly in a serverless context. In: Paasivaara M, Kruchten P (eds). Agile Processes in Software Engineering and Extreme Programming – Workshops. Springer, Cham. pp 141–145
133. Spillner J (2020) Resource management for cloud functions with memory tracing, profiling and autotuning. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 13–18. <https://doi.org/10.1145/3429880.3430094>
134. Spillner J, Mateos C, Monge DA (2018) Faaster, better, cheaper: The prospect of serverless scientific computing and hpc. In: Mocskos E, Nesmachnow S (eds). High Performance Computing. Springer, Cham. pp 154–168
135. Alder F, Asokan N, Kurnikov A, Paverd A, Steiner M (2019) S-faas: Trustworthy and accountable function-as-a-service using intel sgx. In: Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop (CCSW'19). Association for Computing Machinery, New York. pp 185–199. <https://doi.org/10.1145/3338466.3358916>
136. Kuriata A, Illikkal RG (2020) Predictable performance for qos-sensitive, scalable, multi-tenant function-as-a-service deployments. In: Paasivaara M, Kruchten P (eds). Agile Processes in Software Engineering and Extreme Programming – Workshops. Springer, Cham. pp 133–140
137. Mohan A, Sane H, Doshi K, Edupuganti S, Nayak N, Sukhomlinov V (2019) Agile cold starts for scalable serverless. In: Proceedings of the 11th USENIX Conference on Hot Topics in Cloud Computing (HotCloud'19). USENIX Association, USA. p 21
138. Shahrad M, Fonseca R, Goiri I, Chaudhry G, Batum P, Cooke J, Laureano E, Tresness C, Russinovich M, Bianchini R (2020) Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. In: 2020 USENIX Annual Technical Conference (USENIX ATC 20). pp 205–218
139. Obetz M, Das A, Castiglia T, Patterson S, Milanova A (2020) Formalizing event-driven behavior of serverless applications. In: Brogi A, Zimmermann W, Kritikos K (eds). Service-Oriented and Cloud Computing. Springer, Cham. pp 19–29
140. Das A, Imai S, Patterson S, Wittie MP (2020) Performance optimization for edge-cloud serverless platforms via dynamic task placement. In: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). pp 41–50. <https://doi.org/10.1109/CCGrid49817.2020.00-89>
141. Das A, Leaf A, Varela CA, Patterson S (2020) Skedulix: Hybrid cloud scheduling for cost-efficient execution of serverless applications. In: 2020 IEEE 13th International Conference on Cloud Computing (CLOUD). pp 609–618. <https://doi.org/10.1109/CLOUD49709.2020.00090>
142. Obetz M, Patterson S, Milanova A (2019) Static call graph construction in aws lambda serverless applications. In: Proceedings of the 11th USENIX Conference on Hot Topics in Cloud Computing (HotCloud'19). USENIX Association, USA. p 20
143. Lin W, Krintz C, Wolski R, Zhang M, Cai X, Li T, Xu W (2018) Tracking causal order in aws lambda applications. In: 2018 IEEE International Conference on Cloud Engineering (IC2E). pp 50–60. <https://doi.org/10.1109/IC2E.2018.00027>
144. George G, Bakir F, Wolski R, Krintz C (2020) Nanolambda: Implementing functions as a service at all resource scales for the internet of things. In: 2020 IEEE/ACM Symposium on Edge Computing (SEC). pp 220–231. <https://doi.org/10.1109/SEC50012.2020.00035>
145. Zhang M, Krintz C, Wolski R (2020) Stoic: Serverless teleoperable hybrid cloud for machine learning applications on edge device. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp 1–6. <https://doi.org/10.1109/PerComWorkshops48775.2020.9156239>
146. Datta P, Kumar P, Morris T, Grace M, Rahmati A, Bates A (2020) Valve: Securing function workflows on serverless computing platforms. In: Proceedings of The Web Conference 2020 (WWW '20). Association for Computing Machinery, New York. pp 939–950. <https://doi.org/10.1145/3366423.3380173>
147. Sankaran A, Datta P, Bates A (2020) Workflow integration alleviates identity and access management in serverless computing. In: Annual Computer Security Applications Conference (ACSAC '20). Association for Computing Machinery, New York. pp 496–509. <https://doi.org/10.1145/3427228.3427665>
148. Elgamal T (2018) Costless: Optimizing cost of serverless computing through function fusion and placement. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). pp 300–312. <https://doi.org/10.1109/SEC.2018.00029>
149. Dehury CK, Srirama SN, Chhetri TR (2020) Ccodamic: A framework for coherent coordination of data migration and computation platforms. *Futur Gener Comput Syst* 109:1–16. <https://doi.org/10.1016/j.future.2020.03.029>
150. Dehury C, Jakovits P, Srirama SN, Tountopoulos V, Giotis G (2020) Data pipeline architecture for serverless platform. In: Muccini H, Avgeriou P, Buhnova B, Camara J, Caporuscio M, Franzago M, Koziolk A, Scandurra P, Trubiani C, Weyns D, Zdun U (eds). Software Architecture. Springer, Cham. pp 241–246
151. Sarkar S, Wankar R, Srirama SN, Suryadevara NK (2020) Serverless management of sensing systems for fog computing framework. *IEEE Sensors J* 20(3):1564–72. <https://doi.org/10.1109/JSEN.2019.2939182>
152. Malawski M, Gajek A, Zima A, Balis B, Figiela K (2017) Serverless execution of scientific workflows: Experiments with hyperflow, aws lambda and google cloud functions. *Futur Gener Comput Syst*:1–13. <https://doi.org/10.1016/j.future.2017.10.029>
153. Moczurad P, Malawski M (2018) Visual-textual framework for serverless computation: A luna language approach. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). pp 169–74. <https://doi.org/10.1109/UCC-Companion.2018.00052>
154. Pawlik M, Banach P, Malawski M (2020) Adaptation of workflow application scheduling algorithm to serverless infrastructure. In: Schwardmann U, Boehme C, B. Heras D, Cardellini V, Jeannot E, Salis A, Schifanella C, Manumachu RR, Schwamborn D, Ricci L, Sangyoon O, Gruber T, Antonelli L, Scott SL (eds). Euro-Par 2019: Parallel Processing Workshops. Springer, Cham. pp 345–356
155. Akhtar N, Raza A, Ishakian V, Matta I (2020) Cose: Configuring serverless functions using statistical learning. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. pp 129–38. <https://doi.org/10.1109/INFOCOM41043.2020.9155363>
156. Cadden J, Unger T, Awad Y, Dong H, Krieger O, Appavoo J (2020) Seuss: Skip redundant paths to make serverless fast. In: Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys '20). Association for Computing Machinery, New York. <https://doi.org/10.1145/3342195.3392698>
157. Balla D, Maliosz M, Simon C (2020) Open source faas performance aspects. In: 2020 43rd International Conference on Telecommunications and Signal Processing (TSP). pp 358–364. <https://doi.org/10.1109/TSP49548.2020.9163456>
158. Pelle I, Czentye J, Dóka J, Kern A, Geró BP, Sonkoly B (2020) Operating latency sensitive applications on public serverless edge cloud platforms. *IEEE Internet Things J*:1–1. <https://doi.org/10.1109/JIOT.2020.3042428>
159. Balla D, Maliosz M, Simon C, Gehberger D (2020) Tuning runtimes in open source faas. In: Hsu C-H, Kallel S, Lan K-C, Zheng Z (eds). Internet of Vehicles. Technologies and Services Toward Smart Cities. Springer, Cham. pp 250–266
160. Carver B, Zhang J, Wang A, Cheng Y (2019) In search of a fast and efficient serverless dag engine. In: 2019 IEEE/ACM Fourth International Parallel Data Systems Workshop (PDSW). pp 1–10. <https://doi.org/10.1109/PDSW49588.2019.00005>
161. Gadepalli PK, McBride S, Peach G, Cherkasova L, Parmer G (2020) Sledge: A Serverless-First, Light-Weight Wasm Runtime for the Edge. Association for Computing Machinery, New York
162. Gadepalli PK, Peach G, Cherkasova L, Aitken R, Parmer G (2019) Challenges and opportunities for efficient serverless computing at the

- edge. In: 2019 38th Symposium on Reliable Distributed Systems (SRDS). pp 261–2615. <https://doi.org/10.1109/SRDS47363.2019.00036>
163. Somu N, Daw N, Bellur U, Kulkarni P (2020) Panopticon: A comprehensive benchmarking tool for serverless applications. In: 2020 International Conference on COMmunication Systems NETworkS (COMSNETS). pp 144–151. <https://doi.org/10.1109/COMSNETS48256.2020.9027346>
164. Bajaj D, Bharti U, Goel A, Gupta SC (2020) Partial migration for re-architecting a cloud native monolithic application into microservices and faas. In: Badica C, Liatsis P, Kharb L, Chahal D (eds). Information, Communication and Computing Technology. Springer, Singapore. pp 111–124
165. Daw N, Bellur U, Kulkarni P (2020) Xanadu: Mitigating Cascading Cold Starts in Serverless Function Chain Deployments. Association for Computing Machinery, New York
166. HoseinyFarahabady M, Lee YC, Zomaya AY, Tari Z (2017) A qos-aware resource allocation controller for function as a service (faas) platform. In: Maximilien M, Vallecillo A, Wang J, Oriol M (eds). Service-Oriented Computing. Springer, Cham. pp 241–255. [https://doi.org/10.1007/978-3-319-69035-3\\_17](https://doi.org/10.1007/978-3-319-69035-3_17)
167. Kim YK, HoseinyFarahabady MR, Lee YC, Zomaya AY, Jurdak R (2018) Dynamic control of cpu usage in a lambda platform. In: 2018 IEEE International Conference on Cluster Computing (CLUSTER). pp 234–244. <https://doi.org/10.1109/CLUSTER.2018.00041>
168. Kim YK, HoseinyFarahabady MR, Lee YC, Zomaya AY (2020) Automated fine-grained cpu cap control in serverless computing platform. IEEE Transactions on Parallel and Distributed Systems 31(10):2289–2301. <https://doi.org/10.1109/TPDS.2020.2989771>
169. Suresh A, Somashekar G, Varadarajan A, Kakarla VR, Upadhyay H, Gandhi A (2020) Ensure: Efficient scheduling and autonomous resource management in serverless environments. In: 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS). pp 1–10. <https://doi.org/10.1109/ACSOS49614.2020.00020>
170. Suresh A, Gandhi A (2019) Fnsched: An efficient scheduler for serverless functions. In: Proceedings of the 5th International Workshop on Serverless Computing (WoSC '19). Association for Computing Machinery, New York. pp 19–24. <https://doi.org/10.1145/3366623.3368136>
171. Hunhoff E, Irshad S, Thurimella V, Tariq A, Rozner E (2020) Proactive serverless function resource management. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 61–66. <https://doi.org/10.1145/3429880.3430102>
172. Tariq A, Pahl A, Nimmagadda S, Rozner E, Lanka S (2020) Sequoia: Enabling quality-of-service in serverless computing. In: Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20). Association for Computing Machinery, New York. pp 311–327. <https://doi.org/10.1145/3419111.3421306>
173. Crane M, Lin J (2017) An exploration of serverless architectures for information retrieval. In: Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '17). Association for Computing Machinery, New York. pp 241–244. <https://doi.org/10.1145/3121050.3121086>
174. Kim Y, Lin J (2018) Serverless data analytics with flint. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). pp 451–455. <https://doi.org/10.1109/CLOUD.2018.00063>
175. Król M, Psaras I (2017) Nfaas: Named function as a service. In: Proceedings of the 4th ACM Conference on Information-Centric Networking (ICN '17). Association for Computing Machinery, New York. pp 134–144. <https://doi.org/10.1145/3125719.3125727>
176. Pares-Peredo A, Piza-Davila I, Cervantes F (2019) Building and evaluating user network profiles for cybersecurity using serverless architecture. In: 2019 42nd International Conference on Telecommunications and Signal Processing (TSP). pp 164–167. <https://doi.org/10.1109/TSP.2019.8768825>
177. Ivanov V, Smolander K (2018) Implementation of a devops pipeline for serverless applications. In: Kuhrmann M, Schneider K, Pfahl D, Amasaki S, Ciolkowski M, Hebig R, Tell P, Klünder J, Küpper S (eds). Product-Focused Software Process Improvement. Springer, Cham. pp 48–64. [https://doi.org/10.1007/978-3-030-03673-7\\_4](https://doi.org/10.1007/978-3-030-03673-7_4)
178. Chen H, Zhang L-J (2018) Fbaas: Functional blockchain as a service. In: Chen S, Wang H, Zhang L-J (eds). Blockchain – ICBC 2018. Springer, Cham. pp 243–250. [https://doi.org/10.1007/978-3-319-94478-4\\_17](https://doi.org/10.1007/978-3-319-94478-4_17)
179. Lehvä J, Mäkitalo N, Mikkonen T (2018) Case study: Building a serverless messenger chatbot. In: Garrigós I, Wimmer M (eds). Current Trends in Web Engineering. Springer, Cham. pp 75–86. [https://doi.org/10.1007/978-3-319-74433-9\\_6](https://doi.org/10.1007/978-3-319-74433-9_6)
180. Poth A, Schubert N, Riel A (2020) Sustainability efficiency challenges of modern it architectures – a quality model for serverless energy footprint. In: Yilmaz M, Niemann J, Clarke P, Messnarz R (eds). Systems, Software and Services Process Improvement. Springer, Cham. pp 289–301
181. Deese A (2018) Implementation of unsupervised k-means clustering algorithm within amazon web services lambda. In: 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). pp 626–632. <https://doi.org/10.1109/CCGRID.2018.00093>
182. Jangda A, Pinckney D, Brun Y, Guha A (2019) Formal foundations of serverless computing. Proc ACM Program Lang 3(OOPSLA):1–26. <https://doi.org/10.1145/3360575>
183. Back T, Andrikopoulos V (2018) Using a microbenchmark to compare function as a service solutions. In: Kritikos K, Plebani P, de Paoli F (eds). Service-Oriented and Cloud Computing. Springer, Cham. pp 146–160. [https://doi.org/10.1007/978-3-319-99819-0\\_11](https://doi.org/10.1007/978-3-319-99819-0_11)
184. Kritikos K, Skrzypek P (2018) A review of serverless frameworks. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). pp 161–168. <https://doi.org/10.1109/UCC-Companion.2018.00051>
185. Kritikos K, Skrzypek P (2019) Simulation-as-a-service with serverless computing. In: 2019 IEEE World Congress on Services (SERVICES), vol. 2642-939X. pp 200–205. <https://doi.org/10.1109/SERVICES.2019.00056>
186. Ast M, Gaedke M (2017) Self-contained web components through serverless computing. In: Proceedings of the 2nd International Workshop on Serverless Computing (WoSC '17). Association for Computing Machinery, New York. pp 28–33. <https://doi.org/10.1145/3154847.3154849>
187. Trach B, Oleksenko O, Gregor F, Bhatotia P, Fetzter C (2019) Clemmys: Towards secure remote execution in faas. In: Proceedings of the 12th ACM International Conference on Systems and Storages (SYSTOR '19). Association for Computing Machinery, New York. pp 44–54. <https://doi.org/10.1145/3319647.3325835>
188. Aditya P, Akkus IE, Beck A, Chen R, Hilt V, Rimac I, Satzke K, Stein M (2019) Will serverless computing revolutionize nf? Proc IEEE 107(4):667–678. <https://doi.org/10.1109/JPROC.2019.2898101>
189. Baresi L, Filgueira Mendonça D, Garriga M (2017) Empowering low-latency applications through a serverless edge computing architecture. In: De Paoli F, Schulte S, Broch Johnsen E (eds). Service-Oriented and Cloud Computing. Springer, Cham. pp 196–210
190. Tang Y, Yang J (2020) Lambdata: Optimizing serverless computing by making data intents explicit. In: 2020 IEEE 13th International Conference on Cloud Computing (CLOUD). pp 294–303. <https://doi.org/10.1109/CLOUD49709.2020.00049>
191. Ali A, Pincioli R, Yan F, Smirni E (2020) Batch: Machine learning inference serving on serverless platforms with adaptive batching. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '20). pp 1–15
192. Christidis A, Moschogiannis S, Hsu C-H, Davies R (2020) Enabling serverless deployment of large-scale ai workloads. IEEE Access 8:70150–70161. <https://doi.org/10.1109/ACCESS.2020.2985282>
193. Aske A, Zhao X (2018) Supporting multi-provider serverless computing on the edge. In: Proceedings of the 47th International Conference on Parallel Processing Companion (ICPP '18). Association for Computing Machinery, New York. <https://doi.org/10.1145/3229710.3229742>
194. Gunasekaran JR, Thinakaran P, Nachiappan NC, Srivatsa Kannan R, Kandemir MT, Das CR (2020) Characterizing bottlenecks in scheduling microservices on serverless platforms. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). pp 1197–1198. <https://doi.org/10.1109/ICDCS47774.2020.00195>
195. Dziuranski P, Swan J, Indrusiak LS (2018) Value-based manufacturing optimisation in serverless clouds for industry 4.0. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '18). Association for Computing Machinery, New York. pp 1222–1229. <https://doi.org/10.1145/3205455.3205501>
196. Mujezinović A, Ljubović V (2019) Serverless architecture for workflow scheduling with unconstrained execution environment. In: 2019 42nd International Convention on Information and Communication

- Technology, Electronics and Microelectronics (MIPRO). pp 242–246. <https://doi.org/10.23919/MIPRO.2019.8756833>
197. Tricomi G, Giosa D, Merlino G, Romeo O, Longo F (2020) Toward a function-as-a-service framework for genomic analysis. In: 2020 IEEE International Conference on Smart Computing (SMARTCOMP). pp 314–319. <https://doi.org/10.1109/SMARTCOMP50058.2020.00070>
198. Fingler H, Akshintala A, Rossbach CJ (2019) Usetl: Unikernels for serverless extract transform and load why should you settle for less? In: Proceedings of the 10th ACM SIGOPS Asia-Pacific Workshop on Systems (APSys '19). Association for Computing Machinery, New York. pp 23–30. <https://doi.org/10.1145/3343737.3343750>
199. Soltani B, Ghenai A, Zeghib N (2018) Towards distributed containerized serverless architecture in multi cloud environment. *Proc Comput Sci* 134:121–128. <https://doi.org/10.1016/j.procs.2018.07.152>. The 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2018) / The 13th International Conference on Future Networks and Communications (FNC-2018) / Affiliated Workshops
200. Crespo-Cepeda R, Agapito G, Vazquez-Poletti JL, Cannataro M (2019) Challenges and opportunities of amazon serverless lambda services in bioinformatics. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '19). Association for Computing Machinery, New York. pp 663–668. <https://doi.org/10.1145/3307339.3343462>
201. Dash S, Dash DK (2016) Serverless cloud computing framework for smart grid architecture. In: 2016 IEEE 7th Power India International Conference (PILICON). pp 1–6. <https://doi.org/10.1109/POWERI.2016.8077240>
202. Horovitz S, Amos R, Baruch O, Cohen T, Oyar T, Deri A (2019) Faastest - machine learning based cost and performance faas optimization. In: Coppola M, Carlini E, D'Agostino D, Altmann J, Bañares JA (eds). Economics of Grids, Clouds, Systems, and Services. Springer, Cham. pp 171–186. [https://doi.org/10.1007/978-3-030-13342-9\\_15](https://doi.org/10.1007/978-3-030-13342-9_15)
203. Bardsley D, Ryan L, Howard J (2018) Serverless performance and optimization strategies. In: 2018 IEEE International Conference on Smart Cloud (SmartCloud). pp 19–26. <https://doi.org/10.1109/SmartCloud.2018.00012>
204. Jackson D, Clynch G (2018) An investigation of the impact of language runtime on the performance and cost of serverless functions. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). pp 154–160. <https://doi.org/10.1109/UCC-Companion.2018.00050>
205. Kaplunovich A (2019) Tolambda—automatic path to serverless architectures. In: 2019 IEEE/ACM 3rd International Workshop on Refactoring (IWor). pp 1–8. <https://doi.org/10.1109/IWoR.2019.00008>
206. Lee H, Satyam K, Fox G (2018) Evaluation of production serverless computing environments. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). pp 442–450. <https://doi.org/10.1109/CLOUD.2018.00062>
207. Asghar T, Rasool S, Iqbal MU, Qayyum Z, Mian AN, Ubaknma G (2018) Feasibility of serverless cloud services for disaster management information systems. In: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). pp 1054–1057. <https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00175>
208. McGrath G, Brenner PR (2017) Serverless computing: Design, implementation, and performance. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW). pp 405–410. <https://doi.org/10.1109/ICDCSW.2017.36>
209. Christoforou A, Andreou AS (2018) An effective resource management approach in a faas environment. In: ESSCA@UCC. pp 2–8
210. Rajan RAP (2018) Serverless architecture - a revolution in cloud computing. In: 2018 Tenth International Conference on Advanced Computing (ICoAC). pp 88–93. <https://doi.org/10.1109/ICoAC44903.2018.8939081>
211. Rahman MM, Hasibul Hasan M (2019) Serverless architecture for big data analytics. In: 2019 Global Conference for Advancement in Technology (GCAT). pp 1–5. <https://doi.org/10.1109/GCAT47503.2019.8978443>
212. Huber F, Körber N, Mock M (2019) Selena: A serverless energy management system. In: Proceedings of the 5th International Workshop on Serverless Computing (WOSC '19). Association for Computing Machinery, New York. pp 7–12. <https://doi.org/10.1145/3366623.3368134>
213. Aytekin A, Johansson M (2019) Exploiting serverless runtimes for large-scale optimization. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). pp 499–501. <https://doi.org/10.1109/CLOUD.2019.00090>
214. Gabbriellini M, Giallorenzo S, Lanese I, Montesi F, Peressotti M, Zingaro SP (2019) No more, no less: A formal model for serverless computing. In: Riis Nielson H, Tuosto E (eds). Coordination Models and Languages. Springer, Cham. pp 148–157. [https://doi.org/10.1007/978-3-030-22397-7\\_9](https://doi.org/10.1007/978-3-030-22397-7_9)
215. Fasogbon P, You Y, Aksu E (2020) 3d human model creation on a serverless environment. In: 2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). pp 118–122. <https://doi.org/10.1109/ISMAR-Adjunct51615.2020.00044>
216. Dash S, Sodhi R, Sodhi B (2020) A serverless cloud computing framework for real-time appliance-usage recommendation. In: 2020 21st National Power Systems Conference (NPSC). pp 1–6. <https://doi.org/10.1109/NPSC49263.2020.9331847>
217. Thong Tran T, Zhang Y-C, Liao W-T, Lin Y-J, Li M-C, Huang H-S (2020) An autonomous mobile robot system based on serverless computing and edge computing. In: 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS). pp 334–337. <https://doi.org/10.23919/APNOMS50412.2020.9236976>
218. Witte PA, Louboutin M, Modzelewski H, Jones C, Salvage J, Herrmann FJ (2020) An event-driven approach to serverless seismic imaging in the cloud. *IEEE Trans Parallel Distributed Syst* 31(9):2032–2049. <https://doi.org/10.1109/TPDS.2020.2982626>
219. Ghosh BC, Addya SK, Somy NB, Nath SB, Chakraborty S, Ghosh SK (2020) Caching techniques to improve latency in serverless architectures. In: 2020 International Conference on COMMunication Systems NETWORKS (COMSNETS). pp 666–669. <https://doi.org/10.1109/COMSNETS48256.2020.9027427>
220. Quaresma D, Fireman D, Pereira TE (2020) Controlling garbage collection and request admission to improve performance of faas applications. In: 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). pp 175–182. <https://doi.org/10.1109/SBAC-PAD49847.2020.00033>
221. Birman Y, Hindi S, Katz G, Shabtai A (2020) Cost-effective malware detection as a service over serverless cloud using deep reinforcement learning. In: 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). pp 420–429. <https://doi.org/10.1109/CCGrid49817.2020.00-51>
222. Quang T, Peng Y (2020) Device-driven on-demand deployment of serverless computing functions. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp 1–6. <https://doi.org/10.1109/PerComWorkshops48775.2020.9156140>
223. Gunasekaran JR, Thinakaran P, Nachiappan NC, Kandemir MT, Das CR (2020) Fifer: Tackling Resource Underutilization in the Serverless Era. Association for Computing Machinery, New York
224. Müller I, Marroquin R, Alonso G (2020) Lambda: Interactive data analytics on cold data using serverless cloud infrastructure. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20). Association for Computing Machinery, New York. pp 115–130. <https://doi.org/10.1145/3318464.3389758>
225. Chahal D, Ojha R, Ramesh M, Singhal R (2020) Migrating large deep learning models to serverless architecture. In: 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). pp 111–116. <https://doi.org/10.1109/ISSREW51248.2020.00047>
226. Jain P, Munjal Y, Gera J, Gupta P (2020) Performance analysis of various server hosting techniques. *Proc Comput Sci* 173:70–77. <https://doi.org/10.1016/j.procs.2020.06.010>. International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM2020
227. Khatri D, Khatri SK, Mishra D (2020) Potential bottleneck and measuring performance of serverless computing: A literature study. In: 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). pp 161–164. <https://doi.org/10.1109/ICRITO48877.2020.9197837>



228. Kehler S, Zietlow D, Scheffold J, Blochinger W (2020) Self-tuning serverless task farming using proactive elasticity control. *Clust Comput*. <https://doi.org/10.1007/s10586-020-03158-3>
229. Kelly D, Glavin F, Barrett E (2020) Serverless computing: Behind the scenes of major platforms. In: 2020 IEEE 13th International Conference on Cloud Computing (CLOUD). pp 304–312. <https://doi.org/10.1109/CLOUD49709.2020.00050>
230. Ivan C, Vasile R, Dadarlat V (2019) Serverless computing: An investigation of deployment environments for web apis. *Computers* 8(2). <https://doi.org/10.3390/computers8020050>
231. Baresi L, Filgueira Mendonça D (2019) Towards a serverless platform for edge computing. In: 2019 IEEE International Conference on Fog Computing (ICFC). pp 1–10. <https://doi.org/10.1109/ICFC.2019.00008>
232. Hall A, Ramachandran U (2019) An execution model for serverless functions at the edge. In: Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI '19). Association for Computing Machinery, New York. pp 225–236. <https://doi.org/10.1145/3302505.3310084>
233. Akkus IE, Chen R, Rimac I, Stein M, Satzke K, Beck A, Aditya P, Hilt V (2018) Sand: Towards high-performance serverless computing. In: Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '18). USENIX Association, USA. pp 923–935
234. Keshavarzian A, Sharifian S, Seyedin S (2019) Modified deep residual network architecture deployed on serverless framework of iot platform based on human activity recognition application. *Futur Gener Comput Syst* 101:14–28. <https://doi.org/10.1016/j.future.2019.06.009>
235. Palade A, Kazmi A, Clarke S (2019) An evaluation of open source serverless computing frameworks support at the edge. In: 2019 IEEE World Congress on Services (SERVICES), vol. 2642–939X. pp 206–211. <https://doi.org/10.1109/SERVICES.2019.00057>
236. Shahrad M, Balkind J, Wentzlaff D (2019) Architectural implications of function-as-a-service computing. In: Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '52). Association for Computing Machinery, New York. pp 1063–1075. <https://doi.org/10.1145/3352460.3358296>
237. Cicconetti C, Conti M, Passarella A (2020) A decentralized framework for serverless edge computing in the internet of things. *IEEE Trans Netw Serv Manag*:1–1. <https://doi.org/10.1109/TNSM.2020.3023305>
238. De Palma G, Giallorenzo S, Mauro J, Zavattaro G (2020) Allocation priority policies for serverless function-execution scheduling optimisation. In: Kafeza E, Benatallah B, Martinelli F, Hacid H, Bouguettaya A, Motahari H (eds). *Service-Oriented Computing*. Springer, Cham. pp 416–430
239. Djemame K, Parker M, Datsev D (2020) Open-source serverless architectures: an evaluation of apache openwhisk. In: 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC). pp 329–335. <https://doi.org/10.1109/UCC48980.2020.00052>
240. Chadha M, Jindal A, Gerndt M (2020) Towards federated learning using faas fabric. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 49–54. <https://doi.org/10.1145/3429880.3430100>
241. Cheng B, Fuerst J, Solmaz G, Sanada T (2019) Fog function: Serverless fog computing for data intensive iot services. In: 2019 IEEE International Conference on Services Computing (SCC). pp 28–35. <https://doi.org/10.1109/SCC.2019.00018>
242. Mohanty SK, Premsankar G, di Francesco M (2018) An evaluation of open source serverless computing frameworks. In: 2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). pp 115–120. <https://doi.org/10.1109/CloudCom2018.2018.00033>
243. Li J, Kulkarni SG, Ramakrishnan KK, Li D (2019) Understanding open source serverless platforms: Design considerations and performance. In: Proceedings of the 5th International Workshop on Serverless Computing (WOSC '19). Association for Computing Machinery, New York. pp 37–42. <https://doi.org/10.1145/3366623.3368139>
244. Nguyen HD, Zhang C, Xiao Z, Chien AA (2019) Real-time serverless: Enabling application performance guarantees. In: Proceedings of the 5th International Workshop on Serverless Computing (WOSC '19). Association for Computing Machinery, New York. pp 1–6. <https://doi.org/10.1145/3366623.3368133>
245. Saha A, Jindal S (2018) Emars: Efficient management and allocation of resources in serverless. In: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). pp 827–830. <https://doi.org/10.1109/CLOUD.2018.00113>
246. Solaiman K, Adnan MA (2020) Wlec: A not so cold architecture to mitigate cold start problem in serverless computing. In: 2020 IEEE International Conference on Cloud Engineering (IC2E). pp 144–153. <https://doi.org/10.1109/IC2E48712.2020.00022>
247. Chan A, Wang K-TA, Kumar V (2019) BalloonJVM: Dynamically Resizable Heap for FaaS. In: CLOUD COMPUTING 2019: The Tenth International Conference on Cloud Computing, GRIDs, and Virtualization. pp 99–104
248. Handoyo E, Arfan M, Soetrisno YAA, Somantri M, Sofwan A, Sinuraya EW (2018) Ticketing chatbot service using serverless nlp technology. In: 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). pp 325–330. <https://doi.org/10.1109/ICITACEE.2018.8576921>
249. Zhang M, Zhu Y, Zhang C, Liu J (2019) Video processing with serverless computing: A measurement study. In: Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '19). Association for Computing Machinery, New York. pp 61–66. <https://doi.org/10.1145/3304112.3325608>
250. Fan D, He D (2020) A scheduler for serverless framework base on kubernetes. In: Proceedings of the 2020 4th High Performance Computing and Cluster Technologies Conference; 2020 3rd International Conference on Big Data and Artificial Intelligence (HPCCT; BDAI 2020). Association for Computing Machinery, New York. pp 229–232. <https://doi.org/10.1145/3409501.3409503>
251. Thomas S, Ao L, Voelker GM, Porter G (2020) Particle: Ephemeral endpoints for serverless networking. In: Proceedings of the 11th ACM Symposium on Cloud Computing (SoCC '20). Association for Computing Machinery, New York. pp 16–29. <https://doi.org/10.1145/3419111.3421275>
252. Gramaglia M, Serrano P, Banchs A, Garcia-Aviles G, Garcia-Saavedra A, Perez R (2020) The case for serverless mobile networking. In: 2020 IFIP Networking Conference (Networking). pp 779–784
253. Danayi A, Sharifian S (2018) Pess-mina: A proactive stochastic task allocation algorithm for faas edge-cloud environments. In: 2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS). pp 27–31. <https://doi.org/10.1109/ICSPIS.2018.8700543>
254. Mendki P (2020) Evaluating webassembly enabled serverless approach for edge computing. In: 2020 IEEE Cloud Summit. pp 161–166. <https://doi.org/10.1109/IEEECloudSummit48914.2020.00031>
255. Chaudhry SR, Palade A, Kazmi A, Clarke S (2020) Improved qos at the edge using serverless computing to deploy virtual network functions. *IEEE Internet Things J* 7(10):10673–10683. <https://doi.org/10.1109/JIOT.2020.3011057>
256. Benedict S (2020) Serverless blockchain-enabled architecture for iot societal applications. *IEEE Trans Comput Soc Syst* 7(5):1146–1158. <https://doi.org/10.1109/TCSS.2020.3008995>
257. Cicconetti C, Conti M, Passarella A (2020) Uncoordinated access to serverless computing in mec systems for iot. *Comput Netw* 172:107184. <https://doi.org/10.1016/j.comnet.2020.107184>
258. Qiang W, Dong Z, Jin H (2018) Se-lambda: Securing privacy-sensitive serverless applications using sgx enclave. In: Beyah R, Chang B, Li Y, Zhu S (eds). *Security and Privacy in Communication Networks*. Springer, Cham. pp 451–470. [https://doi.org/10.1007/978-3-030-01701-9\\_25](https://doi.org/10.1007/978-3-030-01701-9_25)
259. Prasetyadi G, Hantoro UT, Mutiara AB, Muslim A, Refianti R (2019) Heresy: A serverless web application to store compressed and encrypted document in the form of url. In: 2019 Fourth International Conference on Informatics and Computing (ICIC). pp 1–5. <https://doi.org/10.1109/ICIC47613.2019.8985735>
260. Kim B, Heo S, Lee J, Jeong S, Lee Y, Kim H (2020) Compiler-assisted semantic-aware encryption for efficient and secure serverless computing. *IEEE Internet Things J*:1–1. <https://doi.org/10.1109/JIOT.2020.3031550>
261. O'Meara W, Lennon RG (2020) Serverless computing security: Protecting application logic. In: 2020 31st Irish Signals and Systems Conference (ISSC). pp 1–5. <https://doi.org/10.1109/ISSC49989.2020.9180214>
262. Agache A, Brooker M, Iordache A, Liguori A, Neugebauer R, Piwonka P, Popa D-M (2020) Firecracker: Lightweight virtualization for serverless applications. In: 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20). USENIX Association, Santa Clara. pp 419–434



263. Pinto D, Dias JP, Sereno Ferreira H (2018) Dynamic allocation of serverless functions in iot environments. <https://doi.org/10.1109/euc.2018.00008>
264. Mejía A, Marcillo D, Gualotúa M, Gualotúa T (2020) Serverless based control and monitoring for search and rescue robots. In: 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). pp 1–6. <https://doi.org/10.23919/CISTI49556.2020.9140444>
265. Meißner D, Erb B, Kargl F, Tichy M (2018) Retro-λ: An event-sourced platform for serverless applications with retroactive computing support. In: Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems (DEBS '18). Association for Computing Machinery, New York. pp 76–87. <https://doi.org/10.1145/3210284.3210285>
266. Zhang T, Xie D, Li F, Stutsman R (2019) Narrowing the gap between serverless and its state with storage functions. In: Proceedings of the ACM Symposium on Cloud Computing (SoCC '19). Association for Computing Machinery, New York. pp 1–12. <https://doi.org/10.1145/3357223.3362723>
267. Grogan J, Mulready C, McDermott J, Urbanavicius M, Yilmaz M, Abgaz Y, McCarren A, MacMahon ST, Garousi V, Elger P, Clarke P (2020) A multivocal literature review of function-as-a-service (faas) infrastructures and implications for software developers. In: Yilmaz M, Niemann J, Clarke P, Messnarz R (eds). Systems, Software and Services Process Improvement. Springer, Cham. pp 58–75
268. Qin S, Wu H, Wu Y, Yan B, Xu Y, Zhang W (2020) Nuka: A generic engine with millisecond initialization for serverless computing. In: 2020 IEEE International Conference on Joint Cloud Computing. pp 78–85. <https://doi.org/10.1109/JCC49151.2020.00021>
269. Zuk P, Rządca K (2020) Scheduling methods to reduce response latency of function as a service. In: 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). pp 132–140. <https://doi.org/10.1109/SBAC-PAD49847.2020.00028>
270. Shen J, Yu H, Zheng Z, Sun C, Xu M, Wang J (2020) Serpens: A high-performance serverless platform for nfv. In: 2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS). pp 1–10. <https://doi.org/10.1109/IWQoS49365.2020.9213030>
271. Ginzburg S, Freedman MJ (2020) Serverless isn't server-less: Measuring and exploiting resource variability on cloud faas platforms. In: Proceedings of the 2020 Sixth International Workshop on Serverless Computing (WoSC'20). Association for Computing Machinery, New York. pp 43–48. <https://doi.org/10.1145/3429880.3430099>
272. Tankov V, Golubev Y, Bryksin T (2019) Kotless: A serverless framework for kotlin. In: 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp 1110–1113. <https://doi.org/10.1109/ASE.2019.00114>
273. Vahidinia P, Farahani B, Aliee FS (2020) Cold start in serverless computing: Current trends and mitigation strategies. In: 2020 International Conference on Omni-layer Intelligent Systems (COINS). pp 1–7. <https://doi.org/10.1109/COINS49042.2020.9191377>
274. Silva P, Fireman D, Pereira TE (2020) Prebaking Functions to Warm the Serverless Cold Start. Association for Computing Machinery, New York
275. Tan B, Liu H, Rao J, Liao X, Jin H, Zhang Y (2020) Towards lightweight serverless computing via unikernel as a function. In: 2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS). pp 1–10. <https://doi.org/10.1109/IWQoS49365.2020.9213020>
276. Cordasco G, D'Auria M, Negro A, Scarano V, Spagnuolo C (2020) Fly: A domain-specific language for scientific computing on faas. In: Schwardmann U, Boehme C, B. Heras D, Cardellini V, Jeannot E, Salis A, Schifanella C, Manumachu RR, Schwaborn D, Ricci L, Sangyoon O, Gruber T, Antonelli L, Scott SL (eds). Euro-Par 2019: Parallel Processing Workshops. Springer, Cham. pp 531–544
277. Nupponen J, Taibi D (2020) Serverless: What it is, what to do and what not to do. In: 2020 IEEE International Conference on Software Architecture Companion (ICSA-C). pp 49–50. <https://doi.org/10.1109/ICSA-C50368.2020.00016>
278. Reuter A, Back T, Andrikopoulos V (2020) Cost efficiency under mixed serverless and serverful deployments. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). pp 242–245. <https://doi.org/10.1109/SEAA51224.2020.00049>
279. Mahajan K, Figueiredo D, Misra V, Rubenstein D (2019) Optimal pricing for serverless computing. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp 1–6. <https://doi.org/10.1109/GLOBECOM38437.2019.9013156>
280. Lynn T, Rosati P, Lejeune A, Emeakaroha V (2017) A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. In: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). pp 162–169. <https://doi.org/10.1109/CloudCom.2017.15>
281. Enes J, Expósito RR, Touriño J (2020) Real-time resource scaling platform for big data workloads on serverless environments. *Futur Gener Comput Syst* 105:361–379. <https://doi.org/10.1016/j.future.2019.11.037>
282. Elsakrawy M, Bauer M (2020) Faas2f: A framework for defining execution-sla in serverless computing. In: 2020 IEEE Cloud Summit. pp 58–65. <https://doi.org/10.1109/IEEECloudSummit48914.2020.00015>
283. Bhattacharjee A, Chhokra AD, Kang Z, Sun H, Gokhale A, Karsai G (2019) Barista: Efficient and scalable serverless serving system for deep learning prediction services. In: 2019 IEEE International Conference on Cloud Engineering (IC2E). pp 23–33. <https://doi.org/10.1109/IC2E.2019.00-10>
284. Maissen P, Felber P, Schiavoni V (2020) Faasdom: A benchmark suite for serverless computing. In: Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems (DEBS '20). Association for Computing Machinery, New York. pp 73–84. <https://doi.org/10.1145/3401025.3401738>
285. Xu Z, Zhang H, Geng X, Wu Q, Ma H (2019) Adaptive function launching acceleration in serverless computing platforms. In: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS). pp 9–16. <https://doi.org/10.1109/ICPADS47876.2019.00011>
286. Soltani B, Ghenai A, Zeghib N (2018) A migration-based approach to execute long-duration multi-cloud serverless functions. In: Maamri R, Belal F (eds). Proceedings of the 3rd International Conference on Advanced Aspects of Software Engineering, ICAASE 2018, Constantine, Algeria, December 1-2, 2018 (CEUR Workshop Proceedings), vol. 2326. pp 42–50
287. Martins H, Araujo F, da Cunha PR (2020) Benchmarking serverless computing platforms. *J Grid Comput* 18(4):691–709. <https://doi.org/10.1007/s10723-020-09523-1>

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)