Thesis for the Degree of Licentiate of Engineering

Towards Measuring and Understanding Performance in Infrastructure- and Function-as-a-Service Clouds

Joel Scheuner

Division of Software Engineering
Department of Computer Science & Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden, 2020

Towards Measuring and Understanding Performance in Infrastructure- and Function-as-a-Service Clouds

Joel Scheuner

Technical Report
ISSN 1652-876X
Department of Computer Science & Engineering
Division of Software Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2020.

~100% of benchmarks are wrong.
- Brendan Gregg, Senior Performance Architect

Abstract

Context Cloud computing has become the de facto standard for deploying modern software systems, which makes its performance crucial to the efficient functioning of many applications. However, the unabated growth of established cloud services, such as Infrastructure-as-a-Service (IaaS), and the emergence of new services, such as Function-as-a-Service (FaaS), has led to an unprecedented diversity of cloud services with different performance characteristics.

Objective The goal of this licentiate thesis is to measure and understand performance in IaaS and FaaS clouds. My PhD thesis will extend and leverage this understanding to propose solutions for building performance-optimized FaaS cloud applications.

Method To achieve this goal, quantitative and qualitative research methods are used, including experimental research, artifact analysis, and literature review.

Findings The thesis proposes a cloud benchmarking methodology to estimate application performance in IaaS clouds, characterizes typical FaaS applications, identifies gaps in literature on FaaS performance evaluations, and examines the reproducibility of reported FaaS performance experiments. The evaluation of the benchmarking methodology yielded promising results for benchmark-based application performance estimation under selected conditions. Characterizing 89 FaaS applications revealed that they are most commonly used for short-running tasks with low data volume and bursty workloads. The review of 112 FaaS performance studies from academic and industrial sources found a strong focus on a single cloud platform using artificial micro-benchmarks and discovered that the majority of studies do not follow reproducibility principles on cloud experimentation.

Future Work Future work will propose a suite of application performance benchmarks for FaaS, which is instrumental for evaluating candidate solutions towards building performance-optimized FaaS applications.

Keywords

Cloud Computing, Performance, Benchmarking, Infrastructure-as-a-Service, Function-as-a-Service, Serverless

Acknowledgments

First and foremost, I express special thanks to my advisor and long-term mentor Philipp Leitner for his advice, trust, and collaboration during the journey from my undergraduate studies towards this licentiate thesis. His right balance between giving guidance and autonomy in developing my research fosters my growth towards becoming an independent researcher. I also thank my co-supervisor Jan-Philipp Steghöfer for his valuable feedback. Further, I am grateful for the freedom my examiner Robert Feldt gives me in conducting my research.

Thank you to all my colleagues at the Software Engineering Division for shaping a great work environment and engaging in fun social activities. My gratitude also includes the extended ICET-lab team in Zurich and WASP colleagues at Chalmers and from other Swedish universities. I wish to thank my collaborators at SPEC-RG Cloud for many fruitful discussions and I am looking forward to continuing the good work.

I appreciate the support and visits of my family and friends from Switzerland and abroad. I wish to express my deepest gratitude to my wife Yao for her positivity ☼, love, care, and continuos encouragement throughout my work.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and by the Swedish Research Council VR under grant number 2018-04127 (Developer-Targeted Performance Engineering for Immersed Release and Software Engineers).

List of Publications

Appended Publications

This thesis is based on the following publications:

[α] **J. Scheuner**, P. Leitner
"A Cloud Benchmark Suite Combining Micro and Applications Benchmarks"
Companion of the 9th ACM/SPEC International Conference on Performance Engineering (ICPE): 4th Workshop on Quality-Aware DevOps (QUDOS), 2018.
doi:10.1145/3185768.3186286

[β] **J. Scheuner**, P. Leitner
"Estimating Cloud Application Performance Based on Micro-Benchmark Profiling"
Proceedings of the 11th IEEE International Conference on Cloud Computing (CLOUD), 2018.
doi:10.1109/CLOUD.2018.00019

[γ] S. Eismann, **J. Scheuner**, E. v. Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L. Abad, A. Iosup
"Serverless Applications: Why, When, and How?"
Under revision for IEEE Software.

[δ] **J. Scheuner**, P. Leitner
"Function-as-a-Service Performance Evaluation: A Multivocal Literature Review"
Journal of Systems and Software (JSS), 2020.
doi:10.1016/j.jss.2020.110708

[ε] **J. Scheuner**, P. Leitner
"Transpiling Applications into Optimized Serverless Orchestrations"
*Proceedings of the 4th IEEE FAS*W: 2nd Workshop on Hot Topics in Cloud Computing Performance (HotCloudPerf) at ICAC/SASO, 2019.*
doi:10.1109/FAS-W.2019.00031

Other Publications

The following publications were published before or during my PhD studies. However, they are not appended to this thesis because they were published before my PhD studies [a-f], unrelated to the thesis, or overlapping with the thesis content.

An updated list of all my publications is available on my website[1] and Google Scholar profile[2].

[a] **J. Scheuner**, P. Leitner, J. Cito, H. Gall
"Cloud WorkBench – Infrastructure-as-Code Based Cloud Benchmarking"
Proceedings of the 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2014.
doi:10.1109/CloudCom.2014.98

[b] **J. Scheuner**, P. Leitner, J. Cito, H. Gall
"Cloud WorkBench: Benchmarking IaaS Providers based on Infrastructure-as-Code"
Companion of the 24th International Conference on World Wide Web (WWW Demo), 2015.
doi:10.1145/2740908.2742833

[c] P. Leitner, **J. Scheuner**
"Bursting With Possibilities – an Empirical Study of Credit-Based Bursting Cloud Instance Types"
Proceedings of the 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC), 2015.
doi:10.1109/UCC.2015.39

[d] **J. Scheuner**, G. Mazlami, D. Schöni, S. Stephan, A. De Carli, T. Bocek, B. Stiller
"Probr – A Generic and Passive WiFi Tracking System"
Proceedings of the 41st IEEE Conference on Local Computer Networks (LCN), 2016.
doi:10.1109/LCN.2016.30

[e] **J. Scheuner**, G. Mazlami, D. Schöni, S. Stephan, A. De Carli, T. Bocek, B. Stiller
"Probr Demonstration – Visualizing Passive WiFi Data"
Proceedings of the 41st IEEE Conference on Local Computer Networks (LCN Demo), 2016. ***Best Demo Award LCN'16****.*
doi:10.1109/LCN.2016.135

[f] **J. Scheuner**, P. Leitner, J. Cito, H. Gall
"An Approach and Case Study of Cloud Instance Type Selection for Multi-Tier Web Applications"
Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2017.
doi:10.1109/CCGRID.2017.12

[1] `https://joelscheuner.com/`
[2] `https://scholar.google.com/citations?user=EfD_tnUAAAAJ`

[g] C. Laaber, **J. Scheuner**, P. Leitner
"Software microbenchmarking in the cloud. How bad is it really?"
Empirical Software Engineering (EMSE), 2019.
doi:10.1007/s10664-019-09681-1

[h] **J. Scheuner**, P. Leitner
"Performance Benchmarking of Infrastructure-as-a-Service (IaaS) Clouds with Cloud WorkBench"
Companion of the 10th ACM/SPEC International Conference on Performance Engineering (ICPE Tutorial), 2019.
doi:10.1145/3302541.3310294

[i] **J. Scheuner**, P. Leitner
"Tutorial – Performance Benchmarking of Infrastructure-as-a-Service (IaaS) Clouds with Cloud WorkBench"
Proceedings of the 4th IEEE International Workshops on Foundations and Applications of Self Systems (FAS*W) at ICAC/SASO, 2019.*
doi:10.1109/FAS-W.2019.00070

[j] E. v. Eyk, **J. Scheuner**, S. Eismann, C. L. Abad, A. Iosup
"Beyond Microbenchmarks: The SPEC-RG Vision for a Comprehensive Serverless Benchmark"
Companion of the 11th ACM/SPEC International Conference on Performance Engineering (ICPE): 3rd Workshop on Hot Topics in Cloud Computing Performance (HotCloudPerf), 2020.
doi:10.1145/3375555.3384381

[k] S. Eismann, **J. Scheuner**, E. v. Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L Abad, A. Iosup
"A Review of Serverless Use Cases and their Characteristics"
SPEC-RG-2020-5 Technical Report, 2020. Endorsed by SPEC-RG but not formally peer reviewed.

Research Contribution

Following the Contributor Roles Taxonomy (CrediT)[3]:

For papers α, β, δ and ε, I was the main contributor for Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing - Original Draft, Writing - Review & Editing, Visualization.

Paper γ was the result of collaborative research by SPEC-RG Cloud[4] and a bachelor thesis supervised by me. I was highly involved in Conceptualization, Methodology, Validation, and Writing - Review & Editing. Investigation and Data curation was split equally among the authors. The first author contributed Visualization and Writing - Original Draft based on the underlying technical report SPEC-RG-2020-5 [k], where Writing - Original Draft was largely split equally among the authors.

[3] `https://casrai.org/credit/`

[4] `https://research.spec.org/home.html`

Contents

Chapter 1

Synopsis

Cloud computing [1, 2] has become the de facto standard for deploying modern software systems. The established cloud computing paradigm Infrastructure-as-a-Service (IaaS) grows unabated[1] and the emerging paradigm Serverless computing experiences rapid adoption[2]. IaaS can be seen as the core of cloud environments offering low-level computing resources (e.g., CPU processing time or disk space) as self-service, prevalently in the form of virtual machines (VMs). As cloud computing evolves towards higher-level abstractions, it aims to liberate users entirely from operational concerns, such as managing or scaling server infrastructure. Function-as-a-Service (FaaS) offers such a high-level fully-managed service with fine-grained billing to execute event-triggered code snippets (i.e., functions).

The continuing growth of the cloud computing market has led to an unprecedented diversity of cloud services offered in many different configurations with varying performance characteristics. Hence, selecting an appropriate cloud service with an optimal configuration for a performance- and cost-efficient functioning of an application is a non-trivial challenge.

Performance evaluation is a field of research that systematically measures characteristics such as latency or throughput to build an understanding of performance in a given environment. Performance evaluation in IaaS clouds is an established area of research (see Section 1.3.1) but requires new methods for reproducible experimentation and for understanding the relationship between different performance benchmarks (i.e., performance tests). In contrast, FaaS performance evaluation (see Section 1.3.2) is a much younger but very active area of research that lacks a consolidated understanding. Therefore, this thesis formulates the following goal:

Goal

> My licentiate thesis aims towards measuring and understanding performance in IaaS and FaaS clouds. My PhD thesis will extend and leverage this understanding to propose solutions for building performance-optimized FaaS cloud applications.

[1] `https://aws.amazon.com/ec2/gartner-mq-2019/`

[2] `https://www.marketsandmarkets.com/Market-Reports/serverless-architecture-market-64917099.html`

To achieve this goal, this thesis contributes experimental research to measure and understand performance in IaaS clouds and primary as well as secondary research to understand the landscape of FaaS applications and FaaS performance evaluations.

The remainder of this chapter is organized as follows. Section 1.1 introduces relevant background on cloud computing and the foundations of performance evaluation. Section 1.2 raises and motivates the research questions of this thesis and discusses its scope related to my PhD thesis. Section 1.3 extends the context discussion to related work in the fields of IaaS and FaaS performance evaluation. Section 1.4 summarizes the methodology used to address the research questions. The contributions of the individual papers are summarized and linked to the research questions in Section 1.5. The research questions are then answered in Section 1.6 and discussed in a larger context in Section 1.7. Threats to the validity of the results are discussed in Section 1.8. Section 1.9 outlines future work as part of my PhD thesis and Section 1.10 concludes this thesis.

1.1 Background

This section defines cloud computing, distinguishes FaaS and serverless computing, introduces the foundations of performance evaluation, and distinguishes between micro- and application-level benchmarks.

1.1.1 Cloud Computing

Cloud computing [2–6] is most commonly defined as:

> a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
>
> —The NIST Definition [1]

Cloud computing continues to evolve, moving from low-level generalist services towards more specialized high-level services. Early *Infrastructure-as-a-Service (IaaS)* clouds offer a low-level abstraction of computing resources. These resources are most commonly provided in the form of self-administered *virtual machines (VMs)* where users have near full control of the software stack [7]. Cloud VMs are offered in many different sizes (also called instance types) with different performance and cost characteristics. A prominent example of an IaaS compute service is the Elastic Compute Cloud (EC2) offered by the cloud provider Amazon Web Services (AWS).

As cloud computing matures, new services push towards more fine-grained deployment units of increasingly specialized services as depicted in Figure 1.1. VMs virtualized the hardware of bare metal machines, containers provide virtualization on top of a shared operating system, and Function-as-a-Service (FaaS) offers prepackaged runtimes for high-level application development. FaaS deployment units are small code functions written in programming languages

such as JavaScript or Python. Hence, FaaS allows developers to focus on business logic while abstracting away operational concerns, such as autoscaling VMs.

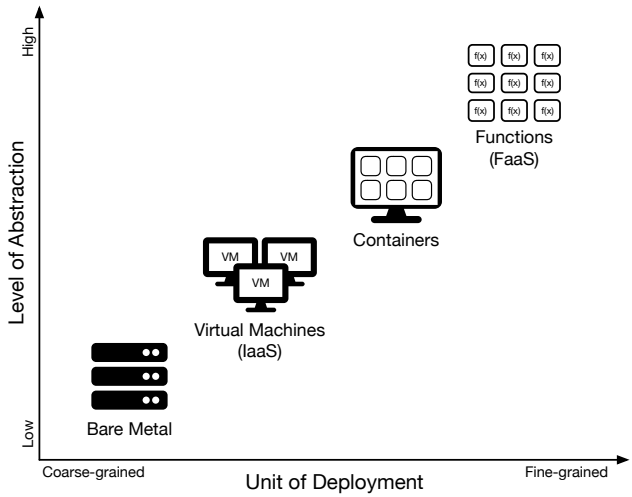

Figure 1.1: Evolution of deployment options

1.1.2 Serverless Computing and Function-as-a-Service

There are no widely accepted definitions for serverless computing and Function-as-a-Service (FaaS). Both terms are often used interchangeably and sometimes even with contradicting interpretations [8]. The term *server-less* (i.e., without *managing* servers) is also considered confusing but widely adopted by academics and practitioners[3]. This thesis adopts an interpretation in line with an accessible introduction to serverless computing [9] and the SPEC cloud group's research vision on FaaS and serverless architectures [10].

> *Serverless computing* is a cloud computing paradigm that aims to liberate users entirely from operational concerns, such as managing or scaling server infrastructure, by offering a fully-managed high-level event-driven service with fine-grained billing.

> *Function-as-a-Service (FaaS)* is one embodiment of serverless computing and is defined through FaaS platforms (e.g., AWS Lambda) executing event-triggered code snippets (i.e., functions).

Figure 1.2 visualizes the relationship between serverless and FaaS and lists example FaaS platforms[4]. This thesis focuses on FaaS but also considers its relevant serverless context given the tight integration. For example, the performance of serverless storage (e.g., AWS S3) can be relevant as part of FaaS applications [11] but not in isolation [12].

[3] `https://martinfowler.com/articles/serverless.html`
[4] `https://landscape.cncf.io/format=serverless`

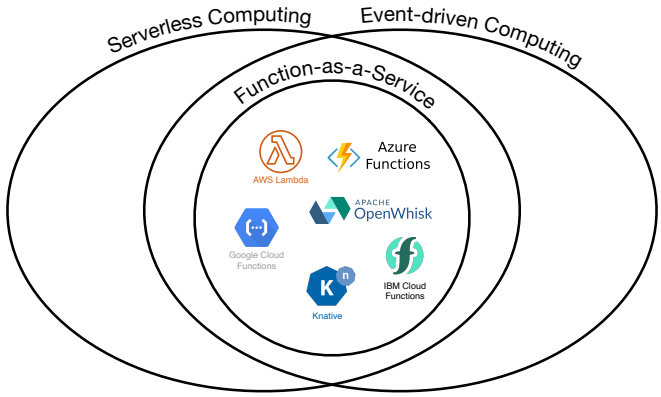

Figure 1.2: Relationship between serverless and FaaS (adapted from [8])

1.1.3 Performance Evaluation

Performance evaluation, also known as performance benchmarking or performance testing, is the process of systematically evaluating performance features (e.g., latency or throughput [13]) of computing resources (e.g., CPU, memory) and applications.

The fundamental performance testing terminology includes: system under test, workload, benchmark, and benchmark suite. A *system under test (SUT)* refers to environments or components that are evaluated according to clearly defined metrics, such as response time. In the context of this thesis, the SUT is typically either a cloud environment (i.e., IaaS or FaaS) or an application within a cloud environment. A *workload* refers to the stimulation that is applied to a SUT to observe a certain effect (e.g., change in performance). This thesis distinguishes between synthetic workloads for micro-benchmarks and realistic workloads for application-benchmarks, which intend to imitate real-world scenarios. A *benchmark* tests performance in a controlled setup by applying a workload to a SUT. A *benchmark suite* groups a set of related benchmarks and defines an execution methodology for combined execution.

Concrete performance features [13], metrics [14], and evaluation methods [15, 16] are cataloged in related work and described within the thesis where relevant.

1.1.4 Micro- and Application-Benchmarks

Figure 1.3 compares two common types of benchmarks, namely micro- and application-benchmarks. *Micro-benchmarks* target a narrow performance aspect (e.g., floating-point CPU performance) with synthetic workloads. These generic benchmarks are not bound to a certain domain (e.g., Web serving) but can provide performance insights that are potentially transferable within certain environments. *Application-benchmarks*, also known as macro-benchmarks, aim to cover the overall performance of real-world application scenarios. Typical metrics are end-to-end response time or throughput. Their results are either specific to a certain application under a given workload or a domain of related applications (e.g., Web serving or scientific computing). Their resource usage profile might be complex and dynamic as they are designed to solve a real-

world task rather than testing a specific resource in isolation. In comparison to micro-benchmarks, application-benchmarks tend to be long-running, complex to configure, and hard to debug due to non-trivial bottleneck analysis caused by heterogeneous resource usage profiles. Examples of both benchmark types are described in Section α.3.3 for IaaS and in Section δ.2 for FaaS.

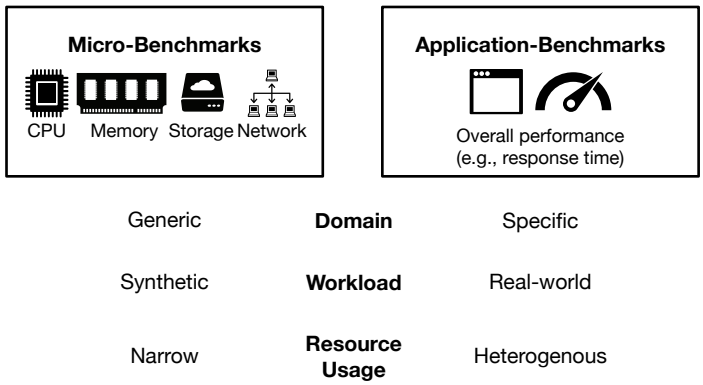

Figure 1.3: Micro- vs application-benchmarks

1.1.5 Reproducibility

"Repeatability and reproducibility are cornerstones of the scientific process" [17] but often neglected in natural [18] and computer [17] science research. Repeatability refers to the extent successive measurements with the same method under *the same conditions* yield the same results [19]. Collberg and Proebsting [17] found that more than 50% of 601 papers from top-rated ACM systems conferences around 2012 lack functional code. Even after spending ample efforts to fix build failures, repeatability was impossible for the results of at least half of these papers. Further, repeatability will likely degrade as Lin and Zhang [20] argued for an understanding as a *process* rather than as an *achievement* due to the fast evolution of modern computational environments. However, reproducibility could still be achieved as it refers to the extent the same results can be achieved with the same method under *changed conditions* of measurements [19].

Following these definitions, repeatability is practically impossible in public cloud environments due to the lack of control over a multi-tenant environment offered by a third-party cloud provider. Therefore, this thesis focuses on technical reproducibility of cloud experimentation, which requires several aspects to ensure an experiment can be repeated with the *same* methodology. A sufficiently detailed experiment description is required but its completeness is often infeasible due to space restrictions (e.g., in academic papers) [21]. Therefore, technical artifacts should be published as an appendix in a usable form [22]. This might include source code, input data (e.g., workloads), and technical descriptions. Access to the same infrastructure is fundamentally given for public clouds but hampered due to their continuous evolution and potentially high costs.

1.2 Research Scope

This section introduces and motivates the research questions of this licentiate thesis and discusses its scope concerning my PhD project and related work.

1.2.1 Research Questions

To addresses the goal of this licentiate thesis, I formulate the following research questions (RQs) covering performance in IaaS (RQ1) and FaaS (RQ2) clouds:

RQ1: *How can performance be measured and evaluated in IaaS clouds?*

Performance evaluation in IaaS clouds is an established field of research (see Section 1.3) and the following sub-questions focus on open challenges. They specifically target measurement methodology with a focus on reproducible IaaS experimentation and the understanding of low-level systems performance in relation to high-level applications. Accordingly, I formulate the follow sub-questions:

RQ1.1: *How can multiple performance benchmarks reproducibly evaluate IaaS cloud performance?*

This question calls for a new cloud benchmarking methodology to systematically combine and execute multiple performance benchmarks. It seeks to improve the reproducibility of cloud experimentation with multiple benchmarks in inherently variable cloud environments.

The ability to systematically evaluate multiple performance benchmarks leads to the follow-up question on how such different performance benchmarks relate to application performance.

RQ1.2: *How suitable are micro-benchmarks to estimate application performance in IaaS clouds?*

By leveraging the execution methodology from RQ1.1., this question aims to explore the potential of generic micro-benchmarks to estimate the performance of specific applications from certain domains to support cloud service selection.

RQ1 targets the performance understanding of low-level IaaS computing resources and RQ2 extends this understanding towards high-level FaaS performance. In contrast to IaaS, performance evaluation in FaaS clouds is a much younger but very active field of research [23]. However, it currently lacks a consolidated view, which motivates the following research question.

RQ2: *What is the current understanding of performance in FaaS clouds?*

This question aims to characterize the landscape of existing work on FaaS applications and their performance to systematically map prior work and guide future research. I divide this question into the following sub-questions related to FaaS applications, FaaS performance characteristics, and the reproducibility of FaaS performance experiments:

RQ2.1: *What are the characteristics of typical FaaS applications?*

A meaningful understanding of FaaS performance requires knowledge about relevant applications, which this question seeks to build.

RQ2.2: *What do existing FaaS performance studies evaluate?*
This question aims to identify and classify the experimental designs of existing FaaS performance studies. Such a consolidated view of existing FaaS performance studies is instrumental to identify gaps in literature and guide future research.

RQ2.3: *How reproducible are existing FaaS performance experiments?*
Reproducibility is an inherently important quality of experimental designs and a common challenge in cloud experimentation, which is performed in inherently variable cloud environments. Hence, this question seeks to assess the reproducibility of existing FaaS performance experiments in terms of compliance with existing guidelines for reproducible cloud experimentation [21].

1.2.2 Research Map

Figure 1.4 visualizes the scope of my licentiate thesis and its relation to my PhD thesis as well as to related work. RQ1 extends related work on IaaS performance evaluation and hereby contributes to the licentiate goals of measuring and understanding performance in IaaS clouds. Specifically, Paper α contributes a IaaS benchmark suite, which Paper β builds upon to propose a new methodology to estimate the performance of cloud application in IaaS clouds. Overall, RQ1 inspires the literature review in Paper δ targeting the active field of performance evaluation in FaaS clouds. However, the lacking knowledge about FaaS applications and their performance requirements in this new field motivated Paper γ, which guides the literature review in Paper δ. Paper γ is a summary of the key findings reported in a more detailed technical report [k] describing the review of existing FaaS applications. Both, Papers γ and δ further guide the work in progress (WIP) on a FaaS application performance benchmark, as envisioned by Paper j. Finally, Paper ε describes a vision of performance-optimized FaaS applications by motivating how the performance understanding from this licentiate thesis could be leverage in future work.

1.3 Related Work

This section discusses related work on IaaS (RQ1) and FaaS (RQ2) performance evaluation.

1.3.1 IaaS Performance Evaluation

Performance evaluation in IaaS cloud environments has a 13-year history with first reports [24–26] appearing around 2007. The first reports followed the beta release of Amazon EC2 in 2006[5], which is considered to be the first commercially available IaaS cloud provider. Since then, cloud performance evaluation has become a popular research area with hundreds of papers published on topics such as benchmarking expectations [27, 28], performance metrics [13, 14], benchmarking approaches [15, 16], performance benchmarks [29], performance experiments [30–33], or hardware heterogeneity [34, 35]. Secondary studies

[5] `https://aws.amazon.com/blogs/aws/amazon_ec2_beta/`

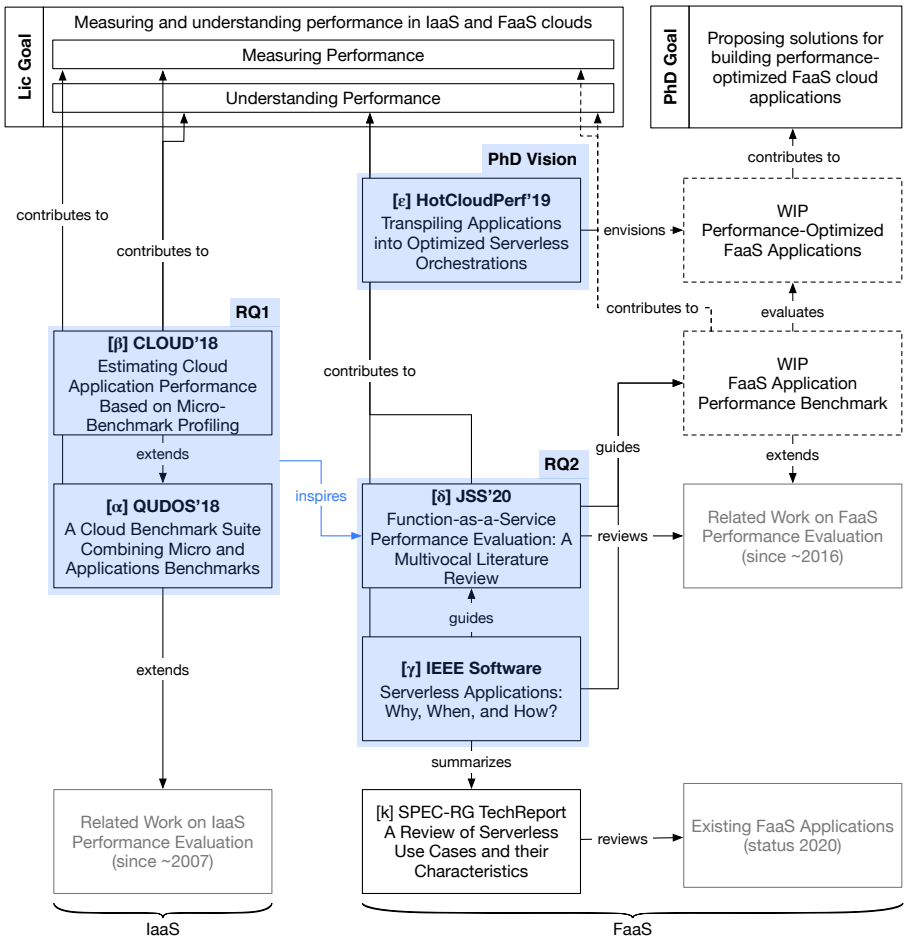

Figure 1.4: Licentiate (Lic) and PhD research map

classified existing research [36] and experimentally validated hypotheses derived by codifying primary studies [37]. Unfortunately, the rapid evolution of cloud systems requires continuous re-evaluation [37] and new methods towards reproducible experimentation in inherently unstable cloud environments [37–39].

1.3.1.1 Cloud Benchmarking Execution Methodology

Existing measurement methodology often makes incorrect assumptions about the underlying system under test when combining multiple performance benchmarks. Abedi and Brecht [40] proposed a new execution methodology called Randomized Multiple Interleaved Trials (RMIT). Figure 1.5 visualizes RMIT with 3 alternatives, which could represent different benchmarks. Single trial and multiple consecutive trials (MCT) are currently the most common methodologies in practice but could lead to erroneous conclusions. Therefore, RMIT should be used to attribute for potential periodic effects in cloud environments beyond the control of experimenters. RMIT was evaluated through simulation based on measurements of micro-benchmarks collected by other researchers [33].

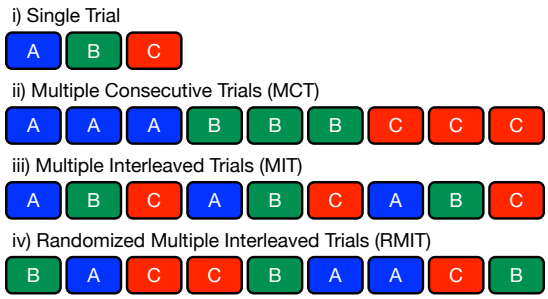

Figure 1.5: Different execution methodologies for 3 alternatives (reproduced from [40]

Several IaaS cloud experiment automation frameworks have been proposed [41–43] but only IBM's Cloud Rapid Experimentation and Analysis Toolkit (CBTOOL)[6] described by Silva et al. [41], Google's PerfKitBenchmarker[7], and my Cloud WorkBench (CWB)[8] framework [44] are still maintained and provide a diverse suite of benchmarks. None of the existing frameworks provide execution methodologies beyond serial trials. Hence, I am not aware of any IaaS benchmark suite that systematically combines multiple benchmarks using a state of the art execution methodology.

1.3.1.2 Cloud Application Performance Prediction

Application performance prediction for optimizing cloud service selection is a common area of research, especially in the context of cloud migration. Initial prediction methods, such as CloudProphet [45], primarily focused on predicting application performance in cloud environments when migrating an application from an on-premise application, for example through trace-and-replay. As cloud offerings started to become more diverse, wholistic methods and tools for cloud rightsizing [46, 47] have been proposed to support cloud migration and optimal service selection. Optimization methods based on micro-benchmarking were proposed and validated for scientific applications [48]. So far, these methods are typically limited to few service types and applications from a single domain. Further, training and validation of existing studies might be negatively impacted by the lack of state of the art execution methodology.

Two of the most related studies were published shortly before and after my paper. Yadwadkar et al. [49] predict the performance of video-encoding and Web serving applications with diverse resource profiles across two cloud providers using hybrid online and offline data collection and modeling. Their profiling benchmarks are limited to their workload requirements, described in insufficient details, and unavailable, neither as code nor dataset. Baughman et al. [50] predict the performance of bioinformatic workflows by combining historical resource traces with online profiling. Neither of the two studies uses interleaved or randomized trials.

[6] https://github.com/ibmcb/cbtool
[7] https://github.com/GoogleCloudPlatform/PerfKitBenchmarker
[8] https://github.com/sealuzh/cloud-workbench

1.3.2 FaaS Performance Evaluation

Performance evaluation in FaaS has a 4-year history with first studies [51, 52] appearing around 2016. The first reports followed the public release of AWS Lambda in 2015[9], which is considered the first FaaS offering by a large public cloud provider. Yussupov et al. [23] indicate that FaaS performance is the most popular area of research in the field of FaaS computing. However, current reports on FaaS performance are disparate originating from different studies executed with different setups and different experimental assumptions. The FaaS community is lacking a consolidated view on the state of research on FaaS performance. To the best of my knowledge, there exists no unified view on FaaS performance and its applications apart from a literature review methodology reporting on preliminary results [53] and two limited collections of FaaS applications.

1.3.2.1 FaaS Application Characteristics

The most extensive curated collection of real-world FaaS applications lists 15 applications[10] and another collection of 13 applications summarizes how serverless is used for 4 common use cases [9]. Cloud providers (e.g., AWS Serverless Application Repository[11]) and FaaS frameworks (e.g., Serverless Framework[12]) publish their collections of FaaS applications but these examples typically rather serve as developer documentation than real-world applications. Other studies addressed developer experience [54] and FaaS patterns [55]. However, the characteristics of individual FaaS applications have not been systematically analyzed by prior work.

1.3.2.2 FaaS Performance Evaluation Landscape

Kuhlenkamp and Werner [53] proposed a methodology for a collaborative literature review on FaaS performance evaluation along with preliminary results. Otherwise, the FaaS performance evaluation landscape has only been discussed as part of limited related work sections in primary studies, most thoroughly by Somu et al. [56].

1.3.2.3 Reproducibility of FaaS Performance Experiments

Reproducibility of performance experiments is an active area of research in IaaS [21, 57] but has not been addressed for FaaS experimentation, apart from preliminary results reported by Kuhlenkamp and Werner [53]. However, their reproducibility score derived from information completeness is limited in scope and coverage.

[9] `https://aws.amazon.com/blogs/compute/aws-lambda-is-generally-available/`
[10] `https://serverlessfirst.com/real-world-serverless-case-studies/`
[11] `https://aws.amazon.com/serverless/serverlessrepo/`
[12] `https://github.com/serverless/examples`

1.4 Research Methodology

This section summarizes the research methodology used to answer the research questions of this thesis. RQ1 was mainly addressed through experimental research methods and RQ2 through a qualitative sample study and a literature review.

1.4.1 Field Experiment

To answer RQ1, an IaaS cloud experiment [15, 16] was conducted as an empirical measurement study in a real cloud environment. According to "the ABC of software engineering research" framework [58], cloud experimentation can be classified as *field experiment* research strategy because the experimental study is conducted in a natural setting (i.e., in a real public cloud environment) but the researcher manipulates some variables (i.e., instance type, benchmark configurations) to observe some effect (i.e., performance metrics).

The experimental research followed a 4-step process depicted in Figure 1.6. First, benchmark design involved a combination of configuring, porting, and implementing several performance benchmarks into a new benchmark suite. Second, benchmark execution consisted of defining experiment plans, scheduling executions, and monitoring multi-week experiments in the real public IaaS cloud environment of AWS EC2. A performance dataset of over 60 000 measurements was collected from over 240 virtual machines across 11 distinct virtual machine types. Third, data pre-processing was required to filter (e.g., skip erroneous executions), re-shape (e.g., transpose or rename), and cleanup (e.g., convert units or replace missing values in a few documented cases) the raw data. Fourth, data analysis included calculating and visualizing summary statistics as well as selecting, optimizing, and evaluating an estimation model.

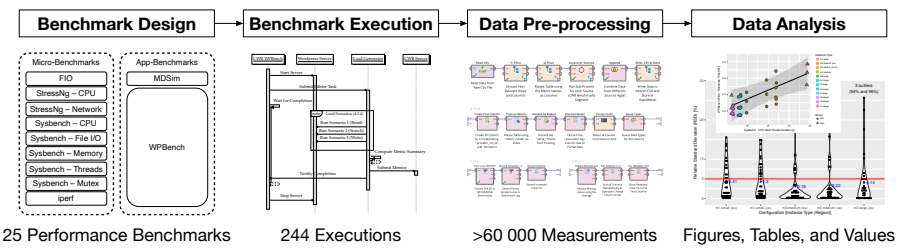

Figure 1.6: Experimental research process

1.4.2 Qualitative Sample Study

To answer RQ2.1, a qualitative sample study was conducted with the main goal to characterize common FaaS applications. The study was conducted in a neutral setting (i.e., desk research) and involved a purely observational analysis of documentation and source code from a broad range of different sources. It can therefore be classified as *sample study* research strategy according to Stol and Fitzgerald [58]. Further, the analysis of documentation and source code qualifies as primary research. The inclusion of academic literature in the broad

data collection might initially hint towards secondary research but the goal was to study FaaS applications and not the contributions of primary studies.

Figure 1.7 summarizes the process of analyzing 89 FaaS applications from four different sources. First, descriptions of FaaS applications were collected from open-source projects, academic literature, industrial literature, and a scientific computing organization. Second, two randomly assigned reviewers out of seven available reviewers characterized each application along 24 characteristics in a structured collaborative review sheet. The characteristics and potential values were defined a priori by the authors and iteratively refined, extended, and generalized during the review process. The initial moderate inter-rater agreement [59] was followed by a discussion and consolidation phase, where all differences between the two reviewers were discussed and resolved. The 6 scientific applications were not publicly available and therefore characterized by a single domain expert, who is either involved in the development of the applications or in direct contact with the development team.

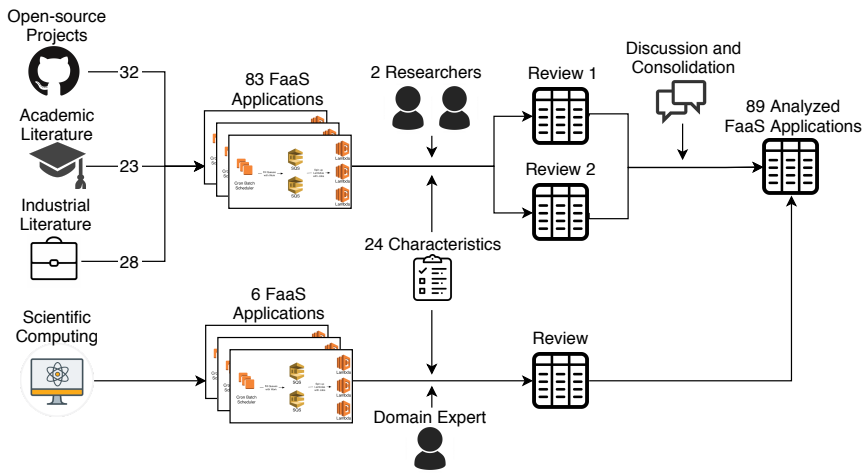

Figure 1.7: Qualitative sample study process

The sampling strategy of FaaS applications is important to achieve a varied sample from different sources, although the qualitative characterization is the primary goal of this study (i.e., following a positivist and reductionist philosophical stance [60]). Following the terminology and guidelines by Baltes and Ralph [61], this study applied different kinds of purposive sampling for the 83 publicly available FaaS applications and convenience sampling for the 6 internal scientific computing applications analyzed by an author employed at the German Aerospace Center. Heterogeneity sampling motivated a roughly balanced selection of open source projects, academic literature (including scientific computing), and industrial literature. Search-based sampling was applied for open source projects through an initial keyword search of the offline GitHub mirror GHTorrent [62] and refined through filtering based on date range, repository activity, repository popularity, and manual selection following inclusion and exclusion criteria. Search-based sampling was applied for academic literature mainly based upon manual selection from the "Serverless Literature Dataset" [63]. Collaborative referral-chain sampling was the main

source for grey literature seeded by case studies reported by cloud providers, an existing article [9], blog posts, forum discussions, and podcasts known to the authors.

1.4.3 Literature Review

To answer RQ2.2 and RQ2.3, I conducted a multivocal literature review (MLR) based on the guidelines from Garousi et al. [64]. Secondary research was suitable to address the lack of a consolidated view on existing FaaS performance evaluation research. Further, the inclusion of grey literature was relevant given the strong industrial interest in FaaS performance and the goal to identify potential mismatches between the academic and industrial perspectives. Figure 1.8 shows how an MLR fits into the landscape of secondary studies and clarifies its scope regarding types of analysis and sources under study. Notice that literature reviews do not fit into the previously discussed "ABC of Software Engineering" framework [58] because the framework exclusively focuses on primary research and knowledge-seeking studies.

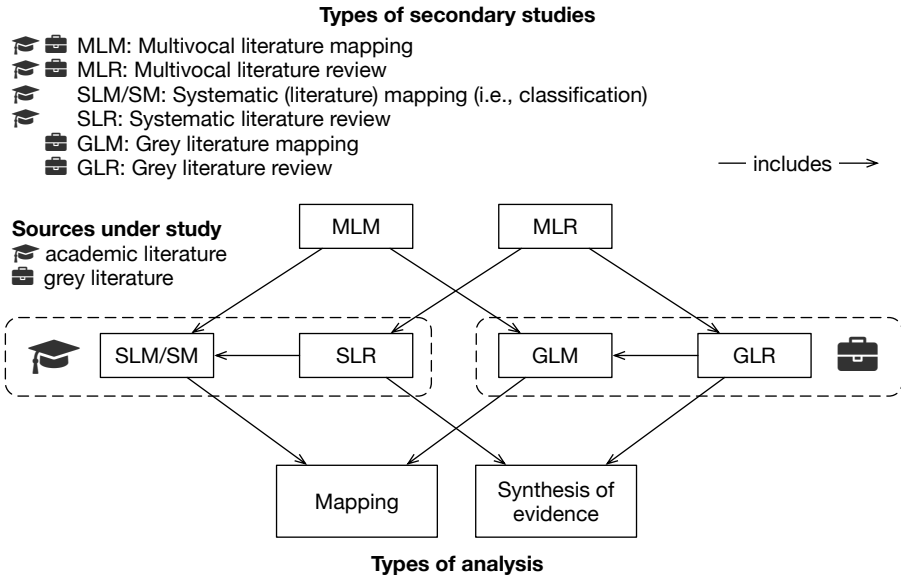

Figure 1.8: Taxonomy of systematic secondary studies (adapted from [64])

Figure 1.9 summarizes the MLR process divided into a part for academic and grey literature. The MLR process identified a total of 112 relevant primary studies. I classified peer-reviewed papers (e.g., papers published in journals, conferences, workshops) as academic literature (i.e., white literature) and other studies (e.g., preprints of unpublished papers, student theses, blog posts) as grey literature. The search process and source selection for academic literature follow a conventional systematic literature review (SLR) process [65]. It was guided through an initial seed of studies [66] discovered through manual search [67] and refined through complementary search strategies, such as alert-based search. The search and selection process for grey literature is based on guidelines for including grey literature [64].

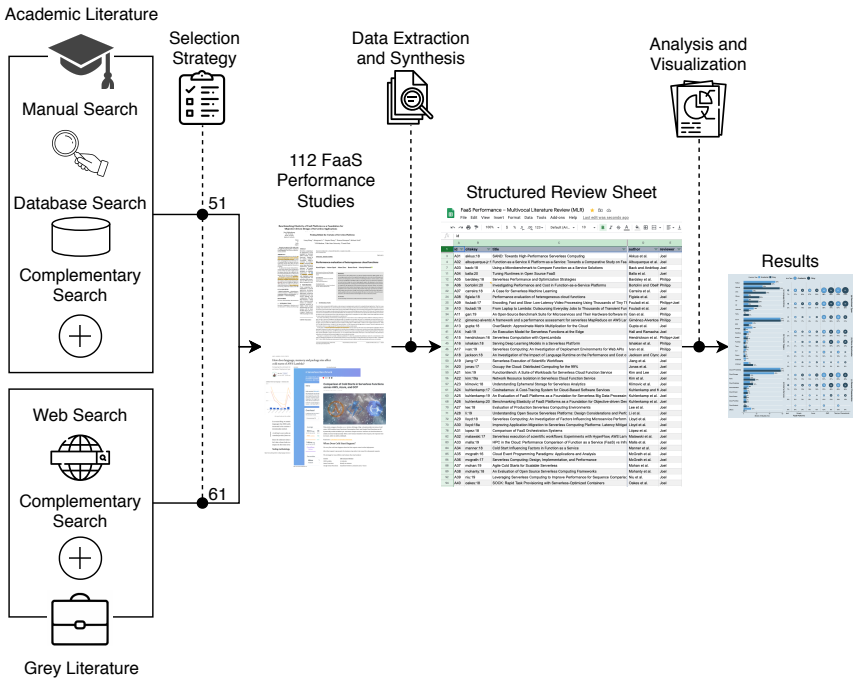

Figure 1.9: Literature review process

1.5 Contributions

This section summarizes the papers, their main contributions this thesis is built on, and their relation to the overarching research questions of this thesis (see Table 1.1). The full papers are appended in the chapters α-ε.

1.5.α A Cloud Benchmark Suite Combining Micro and Applications Benchmarks

Cloud benchmarking literature extensively studied the performance in IaaS clouds using micro- and application-level benchmarks. However, existing work largely focuses on evaluating performance benchmarks in isolation without systematically combining multiple performance benchmarks.

The contribution of Paper α is to fill this gap by presenting an execution methodology that combines micro- and application-benchmarks into a new benchmark suite, integrating this suite into an automated cloud benchmarking framework, and implementing a repeatable execution methodology proposed in related work [40].

Based on cloud benchmarking guidelines [16, 27, 28, 68], relevant benchmarks that cover different cloud resources and application domains were selected, designed, and integrated into the CWB [44] execution framework. The execution of these benchmarks was then automated following the RMIT execution methodology for repeatable experimentation proposed by Abedi and Brecht

Table 1.1: Overview of papers with main contributions

Paper	Venue	Main Contribution	Thesis RQ
α	QUDOS'18	Automated benchmark suite that combines 23 micro- and 2 application-benchmarks	RQ1.1
β	CLOUD'18	Cloud benchmarking methodology for application-benchmark estimation based on micro-benchmark profiling	RQ1.2
γ	IEEE Software (under revision)	Characterization of 89 serverless applications along 24 dimensions regarding motivation, context, and implementation	RQ2.1
δ	JSS'20	Characterization of 112 FaaS performance studies regarding evaluated performance characteristics and configurations	RQ2.2
		Reproducibility assessment of 112 FaaS performance studies based on cloud experimentation guidelines	RQ2.3
ε	HotCloudPerf'19	Vision towards performance-optimized serverless applications	WIP

[40]. The execution methodology was instantiated in the AWS EC2 cloud and the paper presents selected results related to cost-performance efficiency, network bandwidth, and disk utilization.

Paper α layed the methodological and technical foundations for the follow-up study in Paper β. The ability to systematically collect performance measurements for multiple benchmarks raises the question of how their performance relates to each other. In particular, I wanted to explore the potential of generic micro-benchmarks to estimate the performance of specific applications to support cloud service selection.

1.5.β Estimating Cloud Application Performance Based on Micro-Benchmark Profiling

The continuing growth of the cloud computing market has led to an unprecedented diversity of cloud services. To support service selection, micro-benchmarks are commonly used to identify the best performing cloud service. However, it remains unclear how relevant these synthetic micro-benchmarks are for gaining insights into the performance of real-world applications.

Therefore, Paper β contributes a cloud benchmarking methodology for application-benchmark estimation based on micro-benchmark profiling, an evaluation of this methodology in a real IaaS cloud provider, and a performance dataset for micro- and application-benchmarks of over 60 000 measurements from over 240 virtual machines across 11 distinct virtual machine types.

Building upon the benchmark suite from Paper α, these automated benchmarks were repeatedly executed in the AWS EC2 cloud environment and performance measurements were collected from all these benchmark executions. Hereby, relevant metrics (e.g., execution time, response time, latency, throughput, failure rate) were defined and extracted from the detailed output logs of each benchmark trial. Subsequently, offline data pre-processing was required to filter (e.g., skip erroneous executions), re-shape (e.g., transpose or rename), and cleanup (e.g., convert units or replace missing values in a few documented cases) the collected measurements.

The data analysis comprises a prestudy and a main study. A prestudy in line with previous work on cloud benchmarking [31, 37] quantifies the performance variability for equally configured services (i.e., how variable do repeatedly acquired instances of the same instance type perform) because high variability could favor (if correlated) or hamper (if random) meaningful estimates and low variability could facilitate estimation across instance types. Performance variability is quantified as coefficient of variation (CV) across 33 executions for 38 benchmark metrics in five configurations (i.e., different instance types and cloud regions). The main study investigates the suitability of micro-benchmarks for estimating cloud application performance across different instance types in terms of estimation accuracy and micro-benchmark selection. To estimate the application performance, a linear regression model was trained using 38 metrics from 23 micro-benchmarks and evaluated in terms of relative error for two applications from different domains. To select the most relevant estimators, forward feature selection was used to identify the most useful micro-benchmarks and compare them against three common baselines.

Overall, this paper contributes to measuring and understanding performance in low-level IaaS clouds but also helps towards understanding high-level FaaS clouds. In FaaS, providers abstract away operational concerns from the user and thus make them partially inaccessible but still relevant for performance. FaaS platform limitations (e.g., no direct network access, execution time limits) make it impossible to execute the same benchmark suite from this paper in a FaaS environment. However, the concept of using synthetic resource-specific micro-benchmarks to estimate the performance of application-benchmarks inspired by real-world scenarios appears also applicable to FaaS. Unfortunately, the understanding of typical FaaS applications is currently limited and therefore motived further investigation in Paper γ.

The performance experimentation within IaaS clouds (RQ1) highlighted many challenges related to reproducibility and hereby inspired and guided the literature review in Paper δ. The challenges of reusing other application-benchmarks limited the scope of RQ1 and motivated a more systematic analysis of experimental reproducibility for FaaS in Paper δ. The experience of IaaS experimentation also guided the study design of Paper δ. For example, the refinement of the cloud experimentation guideline on open access artifact allowed for a more in-depth discussion on replicating the study design (with provided code) and replicating the (statistical) analyses (with provided dataset). Hence, this research helped to formulate discussions and implications more relevant for future experimenters. It contributed a valuable perspective to the literature review, which is conducted as desk research and inherently limited to explicitly reported results and experiences from primary studies.

1.5.γ Serverless Applications: Why, When, and How?

The emerging cloud computing paradigms Function-as-a-Service and serverless computing are increasingly adopted by industry (as shown by market analyses[13] and surveys[14]) and academics [11, 69–71]. Initial case studies from early adopters indicate significant cost reduction and time-to-market[15] benefits for FaaS applications compared to traditional applications [72]. However, such existing reports are scattered and unstructured. The FaaS community lacks an understanding of typical FaaS applications, which is crucial for designing relevant performance benchmarks.

Therefore, Paper γ characterizes 89 serverless applications along 24 dimensions regarding motivation, context, and implementation to answer questions such as: Why do so many companies adopt serverless?, When are serverless applications well-suited?, and How are serverless applications currently implemented?

Notice that this study focuses on FaaS applications and their serverless context but is framed as *serverless applications* in the appended manuscript. One reason is the aim to target a primarily industrial readership, where FaaS and serverless are often used interchangeably but serverless appears to be known more widespread[16]. Similarly, the term Backend-as-a-Service (BaaS) solutions refers to external services following the serverless paradigm (e.g., AWS S3 for blob storage).

The collection and characterization of existing FaaS applications follow a structured collaborative review process. Descriptions of FaaS applications were collected from diverse sources including open source projects, academic literature, industrial literature, and a scientific computing organization. Each application was either reviewed by two researchers followed by a discussion and consolidation of all disagreements or by a single domain expert for the 6 scientific applications unavailable to the public. A detailed description of the study design is available in the accompanying technical report [73].

The results and insights gained during the review process guided the study design of Paper δ and thus helped to appropriately consolidate existing research on FaaS performance evaluation. For example, the high adoption of integrating external services (e.g., data stores or message queues) in FaaS applications motivated to capture their usage and further analyzing trigger types. The results also emphasize the importance of performance with about 20% of the analyzed FaaS applications explicitly mentioning improved performance as motivation for adopting FaaS. Further, more than 60% of the applications have latency requirements for at least parts of the application, about 40% experience high traffic intensity, and more than 80% exhibit bursty workloads. Cost savings is the most common ($\approx$50%) motivator for FaaS adoption and can also be linked to performance because execution time directly translates into costs with the fine-grained pay-per-use billing model.

[13] `https://www.marketsandmarkets.com/Market-Reports/function-as-a-service-market-127202409.html`

[14] `https://www.oreilly.com/radar/oreilly-serverless-survey-2019-concerns-what-works-and-what-to-expect/`

[15] `https://medium.com/lego-engineering/accelerating-with-serverless-625da076964b`

[16] `https://martinfowler.com/articles/serverless.html`

1.5.δ Function-as-a-Service Performance Evaluation: A Multivocal Literature Review

While performance benchmarking in IaaS clouds has been an active research topic for over a decade (starting 2008), performance benchmarking in FaaS environments is a more recent trend (starting 2015) and lacks a consolidated view on the state of research on FaaS performance. Paper δ fills this gap by conducting the first systematic and comprehensive literature review on FaaS performance evaluation studies from academic and grey literature. It maps the landscape of existing isolated FaaS performance studies, identifies gaps in current research, and systematically investigates their reproducibility based on principles for reproducible performance evaluation [21].

The literature review was designed based on guidelines for systematic literature reviews [65] and multivocal literature reviews [64]. A total of 112 studies were selected from academic (51) and grey (61) literature. The analysis visualizes, describes, and discusses results related to publication trends, benchmarked platforms, evaluated performance characteristics, used platform configurations, and reproducibility of experiments. The paper also highlights and discusses notable differences between academic and grey literature studies.

The implications and gaps in literature identified in this paper directly aim to guide future work on FaaS performance evaluation. Conceptually, the identified gaps in literature from this paper together with the gained understanding of FaaS applications from Paper γ steer future research to address novel and relevant problems. Practically, the datasets of the Papers γ and δ provide a valuable resource for discovering relevant applications and implementations towards a comprehensive FaaS benchmark suite.

1.5.ε Transpiling Applications into Optimized Serverless Orchestrations

The empirical work on reproducible performance evaluation contributes to the understanding of runtime performance monitoring, which is a key idea of the feedback-driven self-adaptive system for building performance-optimized FaaS applications as envisioned in Paper ε. This paper presents the vision of work in progress towards performance-optimized FaaS applications where developers are liberated from conforming to particular forms of deployment units, such as individual functions, and FaaS applications are automatically and dynamically transpiled into a set of individually deployed functions. Such an approach could enable a broader range of serverless applications, lead to more flexible cost-performance trade-off decisions, and increase developer productivity by providing a unified source code view.

1.6 Results

This section answers the research questions raised in Section 1.2.

1.6.1 RQ1: IaaS Performance Evaluation

RQ1: *How can performance be measured and evaluated in IaaS clouds?*

A new benchmark suite and its experimental evaluation demonstrated how to systematically measure IaaS cloud performance by combining multiple performance benchmarks and understanding the connection between micro- and application-level benchmarks. The follow sub-questions provide more detailed answers regarding reproducible measurements (RQ1.1) and understanding performance (RQ1.2):

1.6.1.1 RQ1.1: IaaS Benchmark Suite

RQ1.1: *How can multiple performance benchmarks reproducibly evaluate IaaS cloud performance?*

Main findings: A new IaaS benchmark suite demonstrated that 24 micro- and 2 application-benchmarks can be systematically combined and executed in a fully-automated way. The applied RMIT execution methodology has shown that reproducible execution can be achieved with coefficient of variations (CVs) below 5% for the majority of 38 benchmark metrics across 33 executions in 5 different configurations (i.e., instance types and cloud regions).

Figure 1.10 illustrates the high-level architecture of the proposed benchmark suite. The two most relevant components are the *Benchmark Manager* and the *Cloud VM*. The Benchmark Manager [44] coordinates the entire lifecycle of all benchmark executions. The Cloud VM represents the system under test wherein all benchmarks are automatically installed and configured. The *CWB Client* within a cloud VM steers the execution of the entire benchmark suite following the RMIT execution methodology. The remaining components play a supportive role in resource management, benchmark provisioning, and external load generation.

The selection of benchmarks is motivated by prior use in research and industry. The micro-benchmarks aim for broad resource coverage in the domains computation, I/O, network, and memory but also specifically test individual resources (e.g., dividing I/O into low-level disk I/O and higher-level file I/O with different operation types and sizes). The application benchmarks consist of a Molecular Dynamics Simulation (MDSim) from the scientific computing domain and a Word-Press Benchmark (WPBench) from the Web serving domain.

The results demonstrate the capability of the benchmark suite to achieve low variability (i.e., high precision) for the majority of benchmarks across different VM instances of the same type in multiple configurations. The violin plot in Figure 1.11 compares the variability of 38 benchmark metrics in terms of relative standard deviation (i.e., coefficient of variation) against a 5% relevancy threshold following the definition of a large benchmarking study [37]. The relative standard deviation of each configuration (i.e., the combination of instance type and cloud region) is based on 33 executions from different VM instances of the same type. Each execution is the averaged result of 3 iterations

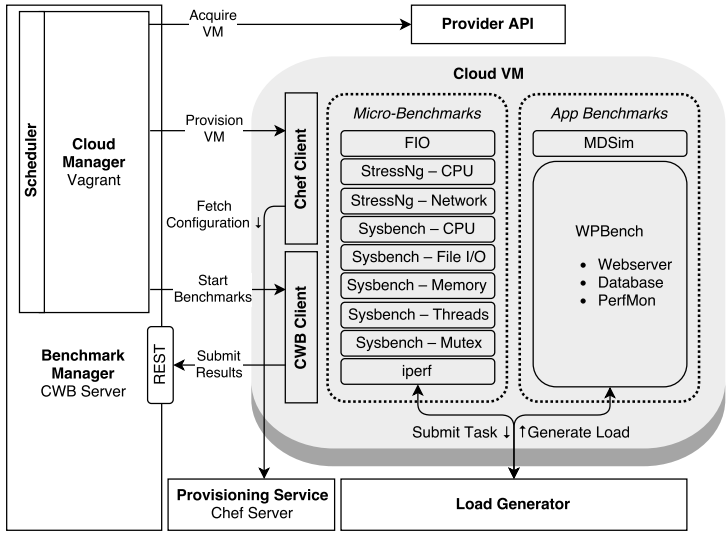

Figure 1.10: Architecture overview of IaaS benchmark suite

(or trials). The results show that the majority of benchmarks are clearly below the 5% threshold for all tested instance types and cloud regions. Similar levels of variability for the same instance types support the ability of the methodology for precise measurements across different regions. The higher variability for the smaller instance type *m1.small* is also in line with the findings of prior work [37, 74, 75].

1.6.1.2 RQ1.2: Application Performance Estimation

RQ1.2: *How suitable are micro-benchmarks to estimate application performance in IaaS clouds?*

> **Main findings:** The evaluation of a linear regression model found that selected micro-benchmarks were able to estimate the duration of a scientific computing application with a relative error of less than 10% and the response time of a Web serving application with a relative error between 10% and 20%. However, it also highlights that benchmarks cannot necessarily be used interchangeably even if they seetest the same resource and benchmark parameters can have a profound impact. Overall, benchmark-based metrics are better in estimating application performance than specification-based metrics.

The proposed methodology uses micro-benchmarks to estimate the performance of an application in previously unseen IaaS computing environments (i.e., on other VM instance types). A suite of systematically combined micro-benchmarks is used to capture a performance footprint of a wide range of VM instance types that are potentially suitable to host an application of interest. The performance of the application is then measured on the smallest and largest candidate instance types to serve as training data. A linear regression model

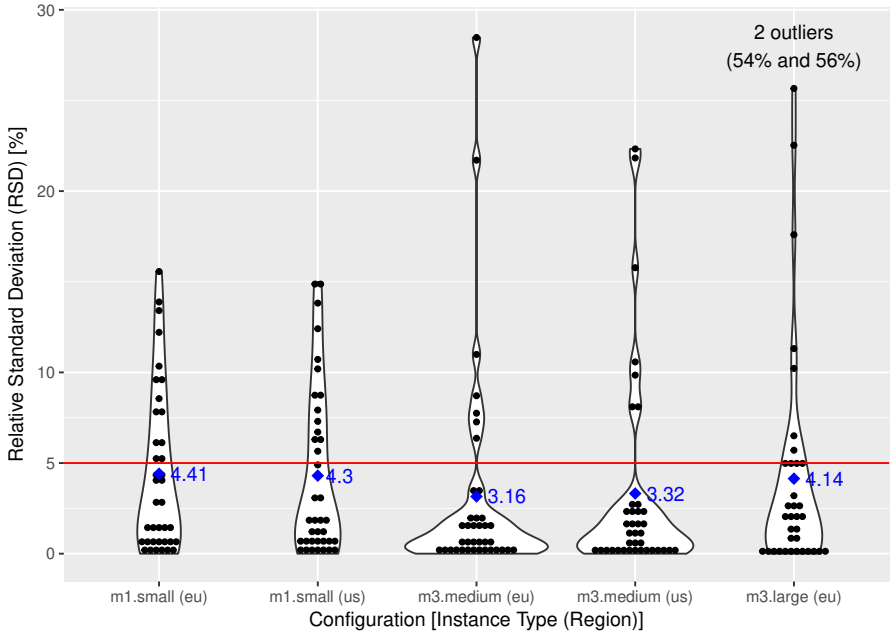

Figure 1.11: Performance variability of benchmarks across different VM instances of the same type

subsequently estimates how the application would perform on a large range of instance types to guide cloud service selection.

The evaluation of the methodology in the AWS EC2 cloud with 11 different instance types showed promising results when using selected micro-benchmarks. Table 1.2 compares the estimation error and degree of correlation for two applications based on two selected benchmarks against common baselines. The estimation accuracy is quantified as average relative error (RE) including standard deviation (±) and the degree of correlation between the predicted and actual values is expressed as coefficient of determination (R^2). The *max RE* estimates the upper bound for the relative error assuming that the smallest instance performs worst and the largest instance performs best. The best benchmark-based estimator (Sysbench CPU Multi-Thread) achieved relative error rates below 10% for the duration of MDSim and between 10% and 20% for the response time of WPBench (omitting the search and write scenario here for brevity). This promising result is much better than a differently configured alternative micro-benchmark estimator (e.g., Sysbench CPU Single-Thread). It also outperforms common baselines, such as the number of virtual CPUs (vCPUs) and a provider-defined unit for describing computational power (ECU). The bad results for cost show that computational power is often not proportionally related to the usage costs.

Table 1.2: Estimators [%] for two applications

	WPBench	**MDSim**
Benchmark	Read Response Time	Duration
Sysbench CPU Multi-Thread		
RE±Range	12.5±7.1	8.2±4.7
R^2	99.2	99.8
Sysbench CPU Single-Thread		
RE±Range	454±520	232±163
R^2	85.1	87.3
Baseline		
vCPUs		
RE±Range	616±607	317±184
R^2	68.0	68.3
ECU		
RE±Range	359±219	206±95
R^2	64.6	65.6
Cost		
RE±Range	663±730	329.3±222
R^2	59.1	57.9
Max Relative Error (RE)	2100	600

1.6.2 RQ2: FaaS Performance Evaluation

RQ2: *What is the current understanding of performance in FaaS clouds?*

Synthetic micro-benchmarks have been studied extensively but the FaaS community lacks a performance understanding of typical production applications. The following sub-questions provide insights into FaaS applications, evaluated FaaS performance aspects, and the reproducibility of FaaS performance experiments:

1.6.2.1 RQ2.1: FaaS Applications

RQ2.1: *What are the characteristics of typical FaaS applications?*

Main findings: The analysis of 89 FaaS applications has shown that FaaS is adopted to save costs for irregular or bursty workloads, to avoid operational concerns, and for the built-in scalability. FaaS applications are most commonly used for short-running tasks with low data volume and bursty workloads but are also frequently used for latency-critical, high-volume core functionality. FaaS applications are mostly implemented on AWS, in either Python or JavaScript, and make heavy use of external services for persistency and coordination functionality.

Figure 1.12 summarizes the most common characteristics of 89 FaaS applications.

The most common motivator for adopting FaaS is cost savings (47%), especially for irregular or bursty workloads. Seemingly-infinite built-in scalability with minimal engineering effort and outsourcing of operational concerns to a provider are both the second most common mentioned by 35% of the 89 FaaS applications. Other reasons were improved performance (19%) and faster time-to-market (13%).

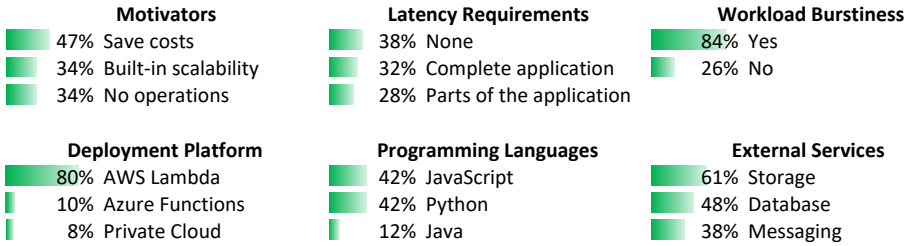

Figure 1.12: Key findings limited to the top 3 values. A single application can have multiple values for motivators, programming languages, and external services.

Latency performance is not relevant for 38% of the FaaS applications. However, 32% of the applications have latency requirements for all functionality, 28% have partial latency requirements, and 2% even mention real-time requirements. Bursty workloads are very typical (84%) for FaaS applications.

AWS Lambda is the most common deployment platform used by 80% of the reviewed 89 applications, followed by Microsoft Azure Functions (10%), and internal hosting in a private cloud (8%). IBM Cloud Functions (7%) and Google Cloud Functions (3%) were less common. Some FaaS applications support multiple deployment platforms and therefore the numbers cumulatively exceed 100%.

JavaScript (42%) and Python (42%) were by far the most popular programming languages. Some applications are also written in Java (12%), C/C++ (11%), or C# (8%), while only a few use Go (5%) or Ruby (2%).

The most common external services provide persistency functionality including blob storage (61%), such as AWS S3, and cloud databases (48%), such as AWS DynomoDB. Different kinds of messaging solutions were seen in 38% of the applications and only 12% integrated no external services.

1.6.2.2 RQ2.2: Existing FaaS Performance Studies

RQ2.2: *What do existing FaaS performance studies evaluate?*

Main findings: The review of 112 performance evaluation studies found that AWS Lambda is the most evaluated FaaS platform (88%), that micro-benchmarks are the most common type of benchmark (75%), and that application benchmarks are prevalently evaluated on a single platform. It also indicates a broad coverage of language runtimes but shows that other platform configurations focus on very few function triggers and external services.

Figure 1.13 summarizes the most commonly evaluated characteristics and configurations of 51 academic and 61 grey literature studies on FaaS performance evaluation.

The most evaluated deployment platforms are AWS Lambda (88%), Microsoft Azure Functions (26%), Google Cloud Functions (23%), IBM Cloud

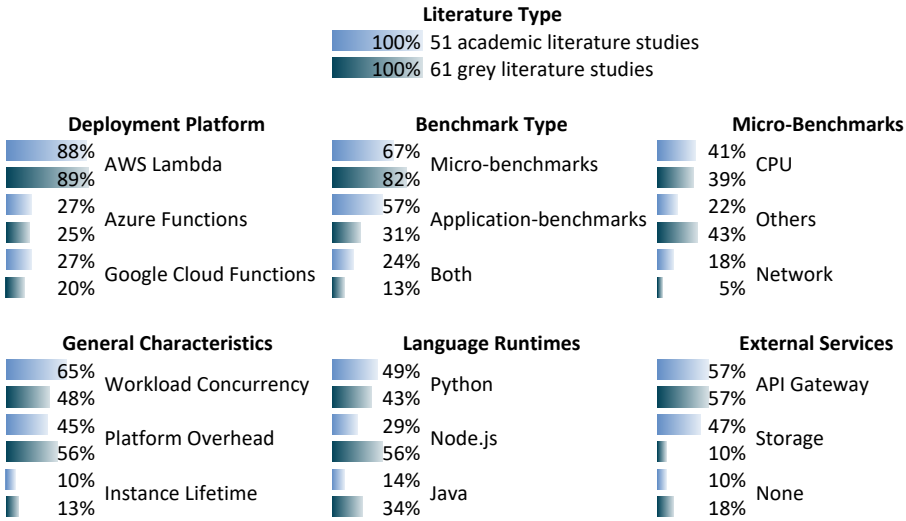

Figure 1.13: Top-3 key findings from 51 academic and 61 grey literature studies on FaaS performance. Multiple values can apply, thus the sum can exceed 100%.

Functions (13%), and self-hosted platforms (14%), predominantly Apache OpenWhisk.

The predominant use of micro-benchmarks in 75% of all studies indicates an over-emphasis on simple easy-to-build benchmarks, compared to application-benchmarks, which are used in 57% of the academic and 31% of the grey literature studies (i.e., overall 18% use both).

Most micro-benchmarks (40%) evaluate CPU performance and show that CPU performance in FaaS systems is indeed proportional to the memory size of the selected function type for certain providers (i.e., AWS, Google). The *Others* category mainly consists of platform overhead and workload concurrency evaluated through micro-benchmarks.

The most evaluated general performance characteristics are FaaS platform overhead (i.e., cold starts) and workload concurrency (i.e., invoking the same function in parallel), both used by about half of the studies.

Language runtimes exhibit a mismatch between academic and industrial sources as Node.js, Java, Go, and C# are evaluated two times more frequently in grey literature than in academic work.

A majority of studies (57%) focuses on HTTP triggers and other trigger types remain largely insufficiently researched. This finding is also reflected in external services with 57% of the studies using an API gateway for implementing HTTP triggers. Cloud storage shows a large discrepancy between academic (47%) and grey (10%) literature studies. Apart from cloud databases (10–15%), external services are used sparingly or not at all (10–18%).

1.6.2.3 RQ2.3: Reproducibility of FaaS Experiments

RQ2.3: *How reproducible are existing FaaS performance experiments?*

Main findings: The review of 112 performance evaluation studies discovered that the majority of studies do not follow principles on reproducible cloud experimentation from prior work [21]. Academic studies tend to satisfy the principles more comprehensively than grey literature, but the data shows no clear trend that academic literature is less susceptible to disregarding the principles.

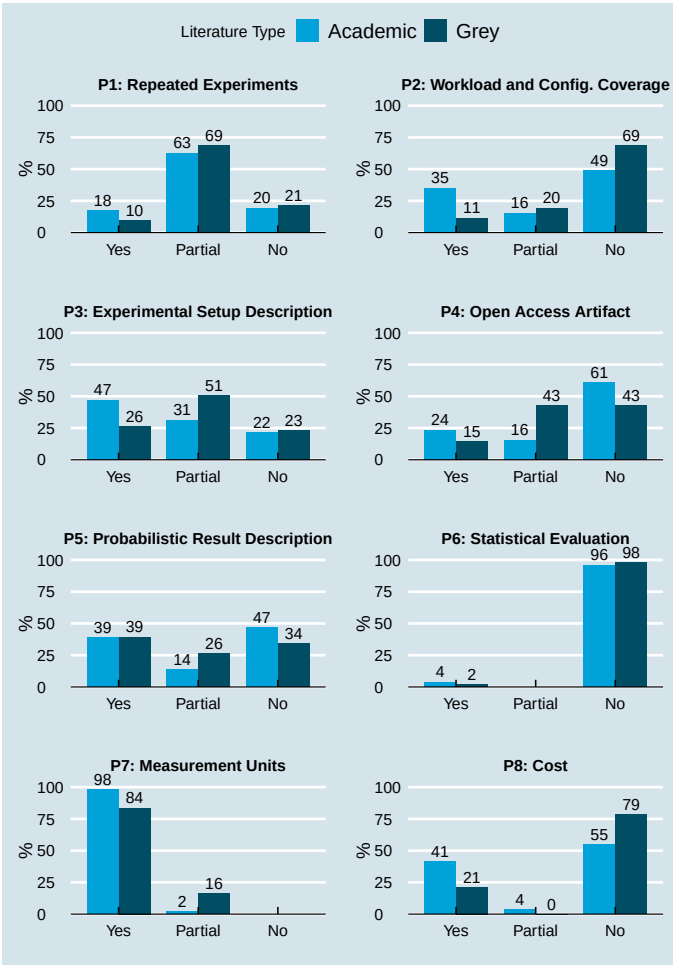

Figure 1.14: Evaluation of reproducibility principles P1–P8 [21] for 51 academic and 61 grey literature studies

Figure 1.14 shows to what extent the reproducibility principles from Papadopoulos et al. [21] are followed by the selected academic and grey literature. Overall, 7 of 8 reproducibility principles are not followed by the majority of the analyzed studies. The first subplot shows that the majority (≈65%) of our selected studies perform some kind of repetition in their experiments, but without justifying or reporting confidence values. About 50% of the academic and 70% of the grey literature studies do not use different workloads and configurations

motivated by real world scenarios. More than half of all studies insufficiently describe their experimental setup. Technical artifacts are unavailable for 61% of the academic and 43% of the grey literature studies. About 40% of all studies appropriately visualize or characterize their empirical performance data, but roughly the same percentage of all studies ignore complex distributions and primarily focus on reporting averages. Almost none of the selected studies perform any statistical evaluations but almost all studies specify measurement units without any major violations. Cost models are missing in 55% of the academic and 79% of grey literature.

The results motivate the following actionable recommendations for future FaaS studies:

P1: Repeated Experiments Explicitly report the number of iterations.

P2: Workload and Configuration Coverage Motivate workloads through industrial use cases.

P3: Experimental Setup Description Report the time of experiment and follow good examples [A26, A8, A34] (see Section δ.5.5–P3).

P4: Open Access Artifact Publish the dataset in addition to the benchmark code.

P5: Probabilistic Result Description Stop reporting mean values exclusively, but use appropriate statistical tools, such as cumulative density functions (CDFs), instead.

P6: Statistical Evaluation Use appropriate statistical tests, such as Wilcoxon rank-sum or overlapping bootstrapped confidence intervals, for stronger conclusions [39].

P7: Measurement Units Include measurement units in all figures.

P8: Cost Report a cost model.

1.7 Discussion

This section discusses the results and implications of the results in the context of related work.

1.7.1 IaaS Performance Evaluation

My instantiation of the new IaaS benchmark suite in a single cloud provider demonstrated the ability to achieve relatively precise (i.e., CVs $<5\%$) results across different instances of the same type. I assume the remaining variability mainly originates from inherently variable cloud environments [37] because several benchmarks achieved almost perfect stability. Further, some outliers indicate inherently unstable benchmarks [39].

In contrast to some prior reports of high performance variability [37, 38], the surprisingly stable results can be partly explained through improved performance stability of the evaluated cloud provider. Nevertheless, the first demonstration of systematically combining micro- and application-benchmarks

using state of the art execution methodologies is a valuable contribution to the IaaS research community towards reproducible experimentation.

The results emphasize the importance of performance benchmarking in IaaS clouds by substantiating the suitability of micro-benchmarks for estimating application performance in comparison to common baselines. An independent study [50] with the same goal published very similar results shortly after my paper for bioinformatic workflows. This supports that the proposed idea is replicable [76] by other researchers with a slightly different methodology. It might also hint towards potential transferability of application estimates within a certain application domain, such as scientific computing. Another independent study [49] with the same goal was published shortly before my paper and reported similar or worse relative errors depending on application, metric, and prediction method. They concluded to achieve better prediction accuracy for non-linear relationships using a random forest prediction model. Their results indicate that the proposed methodology can work across multiple cloud providers. However, I want to highlight that only selected micro-benchmarks were relevant to estimate the performance of a particular application. This limitation motivates future research to focus on detailed profiling and a better understanding of real-world cloud applications as envisioned by Evangelinou et al. [46]. Additionally, I envision that this methodology could be adjusted to be trained directly on monitoring traces from production systems [77] as opposed to dedicated benchmarking efforts.

1.7.2 FaaS Performance Evaluation

Conducted as part of this thesis, the largest systematic analysis of FaaS applications to date contributes a valuable dataset, interesting insights, and encourages community-wide sharing and discussion of FaaS applications. The collection of 89 FaaS applications is a valuable resource of relevant applications for researchers and can guide practitioners on why and when to choose FaaS over alternative paradigms. The results support existing hypotheses (e.g., strong focus on AWS Lambda) but also reveal unexpected (e.g., latency-critical workloads were relatively common) and interesting (e.g., most applications integrate external services) results. The structured characterization and collection of FaaS application hopefully fosters a productive discussion and proves useful for both industry and academia. It goes beyond the existing collections of 10 to 15 real-world applications, which are merely lists with limited discussion and characterization. The most related article by Castro et al. [9] discusses four common use case scenarios in their accessible introduction to FaaS and Serverless computing. Their discussions are rather describing than characterizing and therefore not directly comparable to the results of this thesis.

My results of the first systematic and comprehensive literature review on FaaS performance evaluation studies from academic and grey literature go beyond the existing partial efforts in this area. I analyzed 112 selected studies from over 1500 screened sources, which goes far beyond the most related preliminary work covering 9 out of potential 30 academic studies. Nevertheless, the results of the preliminary study [53] hint towards some of my findings regarding over-emphasis on simple micro-benchmarks and lack of reproducible experimentation in academic studies. However, the preliminary results on

evaluated performance characteristics appear clearly biased due to a very limited sample size and omitted the important industrial perspective, which I covered through the inclusion of grey literature.

1.8 Threats to Validity

This section discusses threats to the validity of the results of this thesis, limitations of the applied research methods, and a summary of mitigation strategies. It is structured based on the four common criteria for validity for empirical research [60]: construct validity, internal validity, external validity, and reliability.

1.8.1 Construct Validity

Construct validity relates to *measuring the right thing*, i.e., the extent a study actually measures what it aims to measure according to the research questions.

For RQ1.1, experimental evaluation demonstrated the feasibility of the proposed methodology and achieved satisfactory results with one possible metric for evaluating the reproducibility of measurements. Other reproducibility aspects were discussed qualitatively but additional quantitative metrics and statistical evaluations could strengthen the reproducibility claim. Benchmark selection and configuration was motivated to cover relevant performance aspects of IaaS cloud performance. However, the large scope of IaaS performance makes it unrealistic to claim complete coverage. For RQ1.2, the main threats concern the selection of appropriate IaaS cloud applications and the choice of error quantification metrics. The selected applications are accepted examples for applications in academic literature [48, 78] from two popular domains but more cloud-native applications [25, 29] could improve the relevance of the application-level workloads. Some related work used different error quantification metrics, such as rankings [48] or normalized service efficiency [46]. My choice of using relative errors is motivated by its usefulness when applying the methodology in practice and was independently adopted by closely related work [49, 50], which was published shortly before and afterward my paper.

For RQ2, construct validity mainly relates to inappropriate selection criteria and a lack of standard language and terminology. To mitigate these threats, the selection criteria were refined based on related work and documented insights from trial classifications. The lack of standard language is a major threat as there exist no established definitions of FaaS and serverless [8]. This threat was mitigated by clarifying and citing selected definitions and providing illustrational examples where applicable.

1.8.2 Internal Validity

Internal validity relates to *measuring right*, i.e., the extent a study measures a causal relationship without interference from external factors.

For RQ1, cloud experimentation is inherently susceptible to confounding factors as a field experiment due to its natural setting [58]. Public clouds cannot be under full control of an experimenter but appropriate execution

methodologies as proposed for RQ1.1 can mitigate this threat. Further mitigation includes careful experimental design based on cloud experimentation guidelines [15, 16] and fully automated experiment execution [44].

For RQ2.1, a qualitative sample study has inherent limitations in measurement precision due to its neutral setting and lack of interactivity (i.e., research must deal with discoverable data *as is*) [58]. To mitigate this threat, each FaaS application was reviewed by two researchers and after initial moderate agreement [73], all differences were discussed and consolidated. The lack of interactive data collection could only be mitigated partially through explorative web search and backward snowballing for discovering new sources. This lead to the explicit labeling of *unknowns* ranging from 0–35% (with two exceptions around 70%) depending on the characteristic. Paper γ excludes these unknowns for brevity and refers to the accompanying technical report [73] for more details and discussion.

For RQ2.2 and RQ2.3, bias in study selection, bias in data extraction, and inappropriate or incomplete database search terms have been identified as the most common threats in literature reviews [79]. To mitigate selection bias, different established search strategies were combined, refined [66], and complemented with targeted strategies (e.g., alert-based search to discover recent studies). Search terms were iteratively refined and motivated in detail (see replication package [80]). Potential inaccuracies in data extraction were mitigated through traceability with over 700 additional comments and a well-defined MLR process based on established guidelines for SLR [65] and MLR [64] studies, methodologically related publications [81], and topically relevant publications [23, 53]. The main threat remains individual researcher bias as the majority of studies were reviewed or validated by a single researcher.

1.8.3 External Validity

External validity relates to *generalizability*, i.e., the extent the results of a study can be transferred to other contexts.

For RQ1, field experimentation inherently lacks statistical generalizability [58]. Thus, I cannot claim generalizability beyond the specific setting studied in two geographically distinct data centers of a single cloud provider across 11 different VM instance types. Although related work also focuses almost exclusively on AWS as a single cloud provider, another study [49] indicated that a similar methodology can also work across multiple cloud providers. This is unsurprising given that most IaaS clouds build upon the same abstractions (i.e., virtualization technology) and individual benchmarks within my suite were previously used across four different cloud providers [37] with the same benchmark manager [44]. In contrast, it is unclear to what extent the results for IaaS are applicable for FaaS. Wang et al. [82] indicated that the underlying hardware infrastructure of AWS Lambda shares the same specifications as VM instance types I have evaluated for answering RQ1.

For RQ2.1, the sampling strategy of the qualitative sample study (Section 1.4.2) motivates a varied mostly purposive sample from different sources. Further, about half of the FaaS applications were classified as deployed in production and the about the same share is open source, which makes the dataset relevant and traceable for publicly documented FaaS applications. However, I

cannot claim generalizability of the results to all FaaS applications, in particular not for private FaaS applications. For RQ2.2 and RQ2.3, the literature review was designed to systematically cover the field of FaaS performance benchmarking for peer-reviewed academic literature (i.e., white literature) and unpublished grey literature including preprints, theses, and articles on the internet. The inclusion of grey literature targets an industrial perspective but is limited to published and indexed content freely available and discoverable on the internet (e.g., excluding paywall articles or internal corporate feasibility studies).

1.8.4 Reliability

Reliability relates to *replicability by others*, i.e., the extent to which the results of a study can be replicated by other researchers.

For RQ1, the experimental design strives for *technical reproducibility* of the data collection and analysis process because the exact reproduction of the measurement results is impossible in this kind of field experimentation due to limited control over the environment [21]. The data collection process leverages Cloud WorkBench (CWB) [44], a web-based cloud experimentation framework purposefully built for technically reproducible performance evaluation in different IaaS clouds and used for other cloud experimentation studies [39, 44, 83, 84]. All performance benchmarks are available as open source software[17] together with extensive documentation, test suites, and automation scripts on how to set up the toolchain and benchmarks. The experimentation framework builds upon appropriate abstractions to isolate technical maintenance changes from more conceptual benchmark definitions but the fast technological evolution might require more corrective maintenance. These mitigation strategies should enable other researchers to conduct the same experiment and collect a new dataset representing the current state of performance. Such a new dataset will be subject to internal changes of the cloud provider, which continuously updates underlying software and hardware infrastructure. Therefore, it is essential to additionally provide the raw dataset and analysis scripts for independent inspection.

The data analysis process strives for replicability [76] based on a documented online replication package[18] providing data and analysis code. The ability to re-run (R^1) the code is hampered by dependencies and could be improved by adopting Docker containerization [85]. Repeatability (R^2) requires repeated code executions to produce the same expected results [76] and was validated by managing interim data with version control. Reproducible (R^3) results require other researchers to be able to re-obtain the same result [76] and is fostered by publicly available data and code under version control but could also be improved by adopting Docker containerization [85]. Reusability (R^4) is partially addressed by documentation but hampered by using a commercial analysis tool. Replicability (R^5) refers to the ability of independent investigators to obtain the same results without re-using the technical artifacts [85] and was partially addressed by re-implementing parts of the analysis in another tool for validation purpose.

[17] `https://github.com/sealuzh/cwb-benchmarks`
[18] `https://github.com/joe4dev/cwb-analysis`

For RQ2, structured review sheets with actionable guidance were used and published in online replication packages [80, 86]. The qualitative sample study alleviated subjective interpretation of the extracted data through multiple reviews from a total of seven reviewers. Bi-lateral and group discussions were an important part of the data consolidation process and captured through systematic spreadsheet commenting and meeting notes but are currently not (yet) publicly available. The literature review mitigated this threat through detailed documentation and traceability annotations.

1.9 Future Work

Future work will firstly extend the current performance understanding for FaaS applications and secondly build on top of this understanding and measurement capabilities to propose solutions for building performance-optimized FaaS applications.

1.9.1 FaaS Application Performance Benchmark

An application-level FaaS performance benchmark suite aims to fill gaps in literature on FaaS performance evaluation identified in this thesis. The FaaS community lacks an application performance benchmark motivated by real-world applications and workloads. This contribution will motivate such an application performance benchmark based on insights from studied FaaS application characteristics. It will further be guided by the insights and actionable recommendations on reproducible FaaS experimentation from this thesis. To its end, I envision that this contribution could lay the foundation for a standardized FaaS benchmarking protocol and implementation. Beyond extending the understanding of FaaS performance, this vision would be a very valuable contribution to the research community for evaluating research prototypes.

To achieve this goal, I am currently leading a long-term effort of the SPEC-RG Cloud[19] working group. This joint work is guided by the insights from this thesis, the expertise of international collaborators, and related work in the field. Hereby, current work in progress highlights challenges specific to evaluating FaaS platforms including:

Performance Requirements Compared to traditional cloud models, such as IaaS, FaaS applications have more stringent performance requirements regarding fast elasticity. Acquiring new VM instances in IaaS yields overheads measured in minutes. In contrast, user-facing FaaS applications typically have latency requirements quantified in milliseconds (see RQ2.1). This motivates the evaluation of platform overhead and workload concurrency (see RQ2.2), which become essential aspects to support for future FaaS benchmarks.

System opaqueness FaaS platforms are opaque by design, attempting to abstract away from the cloud user as much of the operational logic as possible. Despite the benefits of this model, the higher level of abstraction impedes our understanding of what and how internal and external factors influence the

[19] `https://research.spec.org/working-groups/rg-cloud.html`

performance and other characteristics. Therefore, it is valuable to build on top of an understanding of lower-level IaaS cloud (see RQ1) to reason about baseline performance or aspects that cannot be measured (to the same extent) in FaaS clouds due to inherent platform limitations. For example, related work [82] has linked evaluated FaaS infrastructure to a VM instance type I benchmarked as part of answering RQ1. Further, the inability for direct networking in FaaS and the runtime limitations prevent (or hamper) certain network performance characteristics, such as function-to-function network performance. An open question remains to what extent the methodology proposed as part of RQ1 for IaaS cloud is transferable to FaaS clouds.

System heterogeneity The FaaS ecosystem consists of widely heterogeneous systems compared to standardized interfaces for IaaS clouds (i.e., VMs). FaaS platforms have different approaches how functions are built, deployed, scaled, upgraded, and executed. Hence, these are constraints an ecosystem-wide benchmark has to consider and related to active research in the field. For example, Eyk et al. [87] proposed a high-level reference architecture for FaaS platforms and Yussupov et al. [88] introduced a method to automatically assess the portability of FaaS applications.

Complex ecosystems FaaS platforms are, in most cases, not intended as standalone systems. Instead, they provide deep integrations with other cloud services, such as integrations with event sources. To comprehensively evaluate a serverless platform, the performance and implications of these integrations need to be taken into account.

Multi-tenancy and dynamic deployments The short-lived and ephemeral nature of FaaS functions enables cloud providers to dynamically schedule and consolidate the workloads on multiplexed resources. The performance of FaaS platforms varies [82] due to co-located workloads and overall resource demands. These time- and location-related variances need to be considered by a sound FaaS benchmarking methodology. This aspect is very much related to the benchmarking methodology of RQ1 and assessing the reproducibility of FaaS studies (see RQ2.3).

Further details are described in the vision paper "Beyond Microbenchmarks" [89].

1.9.2 Performance-Optimized FaaS Applications

Paper ε presents a vision towards facilitating the development of performance-optimized FaaS applications. The evaluation of the proposed vision requires relevant FaaS applications and can hence leverage the FaaS application benchmark for evaluating the proposed solution. A key motivation for this future work is to make performance insights more actionable in the context of application development through tighter integration of performance aspects into the development cycle. Another avenue for future research could aim to ease the development of multi-function FaaS applications.

1.10 Conclusions

This licentiate thesis extended the existing body of research on measuring and understanding performance in low-level IaaS clouds, established a consolidated understanding of performance in high-level FaaS clouds, and envisioned how this understanding can be leveraged for building performance-optimized FaaS cloud applications. It proposed a benchmark suite combining synthetic micro-benchmarks with real-world application-benchmarks for systematically measuring performance in IaaS clouds. The benchmark suite is part of a cloud benchmarking methodology introduced to estimate application performance in previously unseen compute environments based on performance profiling with micro-benchmarks. An instantiation and evaluation of this methodology yielded promising results that selected micro-benchmarks were able to estimate the performance of two applications from the domains of scientific computing and Web serving. However, the results also highlighted that presumably similar micro-benchmark estimators cannot necessarily be used interchangeably because benchmark parameters can have a profound impact on performance. I conclude that benchmark-based metrics are better estimators for application performance of the tested applications than specification-based metrics (e.g., number of vCPUs, provider-defined unit for computational power), which are currently used as common baselines.

In contrast to IaaS clouds, performance evaluation in high-level FaaS clouds is a more recent but related field of research. The largest analysis of FaaS applications to date identified common performance requirements and other characteristics related to adoption and implementation. In particular, FaaS applications are most commonly used for short-running tasks with low data volume and bursty workloads but are also frequently used for latency-critical, high-volume core functionality. A review of studies on FaaS performance evaluation from academic and industrial sources found that AWS Lambda is the most evaluated FaaS platform, that micro-benchmarks are the most common type of benchmark, and that application benchmarks are prevalently evaluated on a single platform. It also indicated a broad coverage of language runtimes but showed that other platform configurations focus on very few function triggers and external services. Finally, the majority of studies did not follow principles on reproducible cloud experimentation from prior work [21].

The last contribution of this thesis envisioned new solutions towards facilitating the development of performance-optimized FaaS applications through performance feedback-driven application transformation. Towards this goal, current work on an application-level FaaS benchmark suite will enable the evaluation of such envisioned solution prototypes.

Bibliography

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology (NIST), Tech. Rep., 2011. doi:10.6028/NIST.SP.800-145

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[3] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, pp. 50–55, 2008. doi:10.1145/1496091.1496100

[4] D. Hilley, "Cloud computing: A taxonomy of platform and infrastructure-level offerings," Georgia Institute of Technology, Tech. Rep., 2009. [Online]. Available: http://www.cercs.gatech.edu/tech-reports/tr2009/git-cercs-09-13.pdf

[5] S. A. Ahson and M. Ilyas, *Cloud Computing and Software Services: Theory and Techniques*. CRC Press, Inc., 2010.

[6] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and Paradigms*. John Wiley & Sons, 2011.

[7] S. Kächele, C. Spann, F. J. Hauck, and J. Domaschka, "Beyond IaaS and PaaS: An extended cloud taxonomy for computation, storage and networking," in *Proceedings of the 6th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2013, pp. 75–2. doi:10.1109/UCC.2013.28

[8] G. C. Fox, V. Ishakian, V. Muthusamy, and A. Slominski, "Status of serverless computing and function-as-a-service (faas) in industry and research," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1708.08028

[9] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Communications of the ACM*, pp. 44–54, 2019. doi:10.1145/3368454

[10] E. V. Eyk, A. Iosup, S. Seif, and M. Thömmes, "The SPEC Cloud group's research vision on FaaS and serverless architectures," in *Proceedings of the 2nd International Workshop on Serverless Computing (WOSC)*, 2017, pp. 1–4. doi:10.1145/3154847.3154848

[11] E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, and B. Recht, "Occupy the cloud: distributed computing for the 99%," in *Proceedings of the 2017 Symposium on Cloud Computing*, 2017, pp. 445–451. doi:10.1145/3127479.3128601

[12] V. Persico, A. Montieri, and A. Pescapè, "On the network performance of amazon S3 cloud-storage service," in *Proceedings of the 5th IEEE International Conference on Cloud Networking (Cloudnet)*, 2016, pp. 113–118. doi:10.1109/CloudNet.2016.16

[13] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On the conceptualization of performance evaluation of IaaS services," *IEEE Transactions on Services Computing*, pp. 628–41, 2014. doi:10.1109/TSC.2013.39

[14] ——, "On a catalogue of metrics for evaluating commercial cloud services," in *Proceedings of the 13th ACM/IEEE International Conference on Grid Computing (GRID)*, 2012, pp. 164–173. doi:10.1109/Grid.2012.15

[15] Z. Li, L. O'Brien, and H. Zhang, "CEEM: A practical methodology for cloud services evaluation," in *Proceedings of the 9th IEEE World Congress on Services (SERVICES)*, 2013, pp. 44–51. doi:10.1109/SERVICES.2013.73

[16] A. Iosup, R. Prodan, and D. H. J. Epema, "IaaS cloud benchmarking: Approaches, challenges, and experience," in *Cloud Computing for Data-Intensive Applications*, 2014, pp. 83–104. doi:10.1007/978-1-4939-1905-5_4

[17] C. S. Collberg and T. A. Proebsting, "Repeatability in computer systems research," *Communications of the ACM*, pp. 62–69, 2016. doi:10.1145/2812803

[18] M. Baker, "Reproducibility crisis," *Nature*, pp. 353–66, 2016. [Online]. Available: https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970

[19] B. N. Taylor and C. E. Kuyatt, "Guidelines for evaluating and expressing the uncertainty of NIST measurement results," National Institute of Standards and Technology, Tech. Rep., 1994. [Online]. Available: https://emtoolbox.nist.gov/Publications/NISTTechnicalNote1297s.pdf

[20] J. Lin and Q. Zhang, "Reproducibility is a process, not an achievement: The replicability of IR reproducibility experiments," in *Proceedings of the 42nd European Conference on Advances in Information Retrieval IR*, 2020, pp. 43–49. doi:10.1007/978-3-030-45442-5_6

[21] A. V. Papadopoulos, L. Versluis, A. Bauer, N. Herbst, J. von Kistowski, A. Ali-Eldin, C. L. Abad, J. N. Amaral, P. Tuma, and A. Iosup, "Methodological principles for reproducible performance evaluation in cloud computing," *IEEE Transactions on Software Engineering (TSE)*, pp. 93–94, 2019. doi:10.1109/TSE.2019.2927908

[22] D. G. Feitelson, "Experimental computer science: The need for a cultural change," The Hebrew University of Jerusalem, Tech. Rep., 2006.

[23] V. Yussupov, U. Breitenbücher, F. Leymann, and M. Wurster, "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*, 2019, pp. 229–240. doi:10.1145/3344341.3368803

[24] S. L. Garfinkel, "An evaluation of Amazon's grid computing services: EC2, S3, and SQS," Harvard University, Tech. Rep., 2007.

[25] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, S. Patil, A. Fox, and D. Patterson, "Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0," 2008. [Online]. Available: https://pdfs.semanticscholar.org/34dd/c3da70f5b17ae0a73266ad1e4f9ae155811f.pdf

[26] E. Walker, "Benchmarking amazon EC2 for high-performance scientific computing," *Usenix Login*, pp. 18–23, 2008. [Online]. Available: https://www.usenix.org/system/files/login/articles/277-walker.pdf

[27] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow?: Towards a benchmark for the cloud," in *Proceedings of the 2nd International Workshop on Testing Database Systems (DBTest)*, 2009, pp. 9:1–9:6. doi:10.1145/1594156.1594168

[28] E. Folkerts, A. Alexandrov, K. Sachs, A. Iosup, V. Markl, and C. Tosun, "Benchmarking in the cloud: What it should, can, and cannot be," in *Proceedings of the 4th TPC Technology Conference on Selected Topics in Performance Evaluation and Benchmarking (TPCTC)*, 2012, pp. 173–188. doi:10.1007/978-3-642-36727-4_12

[29] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Clearing the clouds: A study of emerging scale-out workloads on modern hardware," *SIGARCH Computer Architecture News*, pp. 37–48, 2012. doi:10.1145/2189750.2150982

[30] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," in *Cloud Computing*, 2009, pp. 115–131. doi:10.1007/978-3-642-12636-9_9

[31] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, pp. 931–945, 2011. doi:10.1109/TPDS.2011.66

[32] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," in *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC)*, 2010, pp. 143–154. doi:10.1145/1807128.1807152

[33] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime measurements in the cloud: Observing, analyzing, and reducing variance," *Proceedings of the VLDB Endowment*, pp. 460–471, 2010. doi:10.14778/1920841.1920902

[34] Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Exploiting hardware heterogeneity within the same instance type of amazon EC2," in *Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Computing (HotCloud)*, 2012. [Online]. Available: https://www.usenix.org/conference/hotcloud12/workshop-program/presentation/ou

[35] Z. Ou, H. Zhuang, A. Lukyanenko, J. K. Nurminen, P. Hui, V. Mazalov, and A. Ylä-Jääski, "Is the same instance type created equal? exploiting heterogeneity of public clouds," *IEEE Transactions on Cloud Computing*, pp. 201–14, 2013. doi:10.1109/TCC.2013.12

[36] Z. Li, H. Zhang, L. O'Brien, R. Cai, and S. Flint, "On evaluating commercial cloud services: A systematic review," *Journal of Systems and Software*, pp. 2371–2393, 2013. doi:10.1016/j.jss.2013.04.021

[37] P. Leitner and J. Cito, "Patterns in the chaos – a study of performance variation and predictability in public IaaS clouds," *ACM Transactions on Internet Technology*, pp. 1–23, 2016. doi:10.1145/2885497

[38] A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production cloud services," in *Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2011, pp. 104–13. doi:10.1109/CCGrid.2011.22

[39] C. Laaber, J. Scheuner, and P. Leitner, "Software microbenchmarking in the cloud. How bad is it really?" *Empirical Software Engineering (EMSE)*, pp. 2469–2508, 2019. doi:10.1007/s10664-019-09681-1

[40] A. Abedi and T. Brecht, "Conducting repeatable experiments in highly variable cloud computing environments," in *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering (ICPE)*, 2017, pp. 287–292. doi:10.1145/3030207.3030229

[41] M. Silva, M. Hines, D. Gallo, Q. Liu, K. D. Ryu, and D. Da Silva, "Cloudbench: Experiment automation for cloud environments," in *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E)*, 2013, pp. 302–11. doi:10.1109/IC2E.2013.33

[42] M. Cunha, N. das Chagas Mendonça, and A. Sampaio, "Cloud crawler: a declarative performance evaluation environment for infrastructure-as-a-service clouds," *Concurrency and Computation: Practice and Experience*, 2017. doi:10.1002/cpe.3825

[43] D. Jayasinghe, G. Swint, S. Malkowski, J. Li, Q. Wang, J. Park, and C. Pu, "Expertus: A generator approach to automate performance testing in IaaS clouds," in *Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD)*, 2012, pp. 115–22. doi:10.1109/CLOUD.2012.98

[44] J. Scheuner, P. Leitner, J. Cito, and H. Gall, "Cloud WorkBench – infrastructure-as-code based cloud benchmarking," in *Proceedings of the 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2014, pp. 246–253. doi:10.1109/CloudCom.2014.98

[45] A. Li, X. Zong, S. Kandula, X. Yang, and M. Zhang, "Cloudprophet: Towards application performance prediction in cloud," in *Proceedings of the ACM SIGCOMM 2011 Conference (SIGCOMM)*, 2011, pp. 426–427. doi:10.1145/2018436.2018502

[46] A. Evangelinou, M. Ciavotta, D. Ardagna, A. Kopaneli, G. Kousiouris, and T. Varvarigou, "Enterprise applications cloud rightsizing through a joint benchmarking and optimization approach," *Future Generation Computer Systems*, pp. 102–114, 2016. doi:10.1016/j.future.2016.11.002

[47] O. Alipourfard, H. H. Liu, J. Chen, S. Venkataraman, M. Yu, and M. Zhang, "Cherrypick: Adaptively unearthing the best cloud configurations for big data analytics," in *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2017, pp. 469–482. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/alipourfard

[48] B. Varghese, O. Akgun, I. Miguel, L. Thai, and A. Barker, "Cloud benchmarking for maximising performance of scientific applications," *IEEE Transactions on Cloud Computing*, pp. 170–182, 2019. doi:10.1109/TCC.2016.2603476

[49] N. J. Yadwadkar, B. Hariharan, J. E. Gonzalez, B. Smith, and R. H. Katz, "Selecting the best VM across multiple public clouds: a data-driven performance modeling approach," in *Proceedings of the Symposium on Cloud Computing (SoCC)*, 2017, pp. 452–465. doi:10.1145/3127479.3131614

[50] M. Baughman, R. Chard, L. T. Ward, J. Pitt, K. Chard, and I. T. Foster, "Profiling and predicting application performance on the cloud," in *Proceedings of the 11th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2018, pp. 21–30. doi:10.1109/UCC.2018.00011

[51] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Serverless Computation with OpenLambda," in *Proceedings of the 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, 2016. [Online]. Available: https://www.usenix.org/conference/hotcloud16/workshop-program/presentation/hendrickson

[52] G. McGrath, J. Short, S. Ennis, B. Judson, and P. Brenner, "Cloud event programming paradigms: Applications and analysis," in *Proceedings of the 9th IEEE International Conference on Cloud Computing (CLOUD)*, 2016, pp. 400–06. doi:10.1109/CLOUD.2016.0060

[53] J. Kuhlenkamp and S. Werner, "Benchmarking FaaS platforms: Call for community participation," in *Companion of the 11th IEEE/ACM UCC: 4th International Workshop on Serverless Computing (WoSC)*, 2018, pp. 189–194. doi:10.1109/UCC-Companion.2018.00055

[54] P. Leitner, E. Wittern, J. Spillner, and W. Hummer, "A mixed-method empirical study of function-as-a-service software development in industrial practice," *Journal of Systems and Software*, pp. 340–359, 2019. doi:10.1016/j.jss.2018.12.013

[55] D. Taibi, N. El Ioini, C. Pahl, and J. R. S. Niederkofler, "Patterns for serverless functions (function-as-a-service): A multivocal literature review," *Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER)*, 2020. doi:10.5220/0009578501810192

[56] N. Somu, N. Daw, U. Bellur, and P. Kulkarni, "Panopticon: A comprehensive benchmarking tool for serverless applications," in *Proceedings of the International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2020, pp. 144–51. doi:10.1109/COMSNETS48256.2020.9027346

[57] A. Uta, A. Custura, D. Duplyakin, I. Jimenez, J. S. Rellermeyer, C. Maltzahn, R. Ricci, and A. Iosup, "Is big data performance reproducible in modern cloud networks?" in *Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2020, pp. 513–527. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/uta

[58] K.-J. Stol and B. Fitzgerald, "The ABC of software engineering research," *ACM Transactions on Software Engineering and Methodology*, pp. 1–51, 2018. doi:10.1145/3241743

[59] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977. doi:10.2307/2529310

[60] S. Easterbrook, J. Singer, M. D. Storey, and D. E. Damian, "Selecting empirical methods for software engineering research," in *Guide to Advanced Empirical Software Engineering*, 2008, pp. 285–311. doi:10.1007/978-1-84800-044-5_11

[61] S. Baltes and P. Ralph, "Sampling in software engineering research: A critical review and guidelines," *CoRR*, 2020. [Online]. Available: https://arxiv.org/abs/2002.07764

[62] G. Gousios, "The GHTorent dataset and tool suite," in *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR)*, 2013, pp. 233–236. doi:10.1109/MSR.2013.6624034

[63] J. Spillner and M. Al-Ameen, "Serverless literature dataset," 2019. doi:10.5281/zenodo.2649001

[64] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, pp. 101–121, 2019. doi:10.1016/j.infsof.2018.09.006

[65] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University, Tech. Rep., 2007. [Online]. Available: http://cdn.elsevier.com/promis_misc/525444systematicreviewsguide.pdf

[66] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Information and Software Technology*, pp. 625–637, 2011. doi:10.1016/j.infsof.2010.12.010

[67] Y. Zacchia Lun, A. D'Innocenzo, F. Smarra, I. Malavolta, and M. D. a. Di Benedetto, "State of the art of cyber-physical systems security: An automatic control perspective," *Journal of Systems and Software*, pp. 174–216, 2019. doi:10.1016/j.jss.2018.12.006

[68] V. Vedam and J. Vemulapati, "Demystifying cloud benchmarking paradigm – an in depth view," in *Proceedings of the 36th IEEE Computer Software and Applications Conference (COMPSAC)*, 2012, pp. 416–21. doi:10.1109/COMPSAC.2012.61

[69] S. Fouladi, R. S. Wahby, B. Shacklett, K. Balasubramaniam, W. Zeng, R. Bhalerao, A. Sivaraman, G. Porter, and K. Winstein, "Encoding, fast and slow: Low-latency video processing using thousands of tiny threads," in *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2017, pp. 363–376. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/fouladi

[70] S. Fouladi, F. Romero, D. Iter, Q. Li, S. Chatterjee, C. Kozyrakis, M. Zaharia, and K. Winstein, "From laptop to lambda: Outsourcing everyday jobs to thousands of transient functional containers," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, 2019, pp. 475–488. [Online]. Available: https://www.usenix.org/conference/atc19/presentation/fouladi

[71] S. Bebortta, S. K. Das, M. Kandpal, R. K. Barik, and H. Dubey, "Geospatial serverless computing: Architectures, tools and future directions," *ISPRS International Journal of Geo-Information*, p. 311, 2020. doi:10.3390/ijgi9050311

[72] G. Adzic and R. Chatley, "Serverless computing: Economic and architectural impact," in *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering (FSE)*, 2017, pp. 884–889. doi:10.1145/3106237.3117767

[73] S. Eismann, J. Scheuner, E. van Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L. Abad, and A. Iosup, "A review of serverless use cases and their characteristics," SPEC RG Cloud Working Group, Tech. Rep., 2020. [Online]. Available: https://research.spec.org/fileadmin/user_upload/documents/rg_cloud/endorsed_publications/SPEC_RG_2020_Serverless_Usecases.pdf

[74] G. Wang and T. S. E. Ng, "The impact of virtualization on network performance of amazon EC2 data center," in *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 1163–1171. doi:10.1109/INFCOM.2010.5461931

[75] L. Kotthoff, "Reliability of computational experiments on virtualised hardware," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 33–9, 2014. doi:10.1080/0952813X.2013.784812

[76] F. C. Y. Benureau and N. P. Rougier, "Re-run, repeat, reproduce, reuse, replicate: Transforming code into scientific contributions," *Frontiers in Neuroinformatics*, p. 69, 2018. doi:10.3389/fninf.2017.00069

[77] P. Bodik, R. Griffith, C. Sutton, A. Fox, M. I. Jordan, and D. A. Patterson, "Automatic exploration of datacenter performance regimes," in *Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds*, 2009, pp. 1–6. doi:10.1145/1555271.1555273

[78] A. H. Borhani, P. Leitner, B. Lee, X. Li, and T. Hung, "WPress: An application-driven performance benchmark for cloud-based virtual machines," in *Proceedings of the 18th IEEE International Enterprise Distributed Object Computing Conference (EDOC)*, 2014, pp. 101–109. doi:10.1109/EDOC.2014.23

[79] X. Zhou, Y. Jin, H. Zhang, S. Li, and X. Huang, "A map of threats to validity of systematic literature reviews in software engineering," in *Proceedings of the 23rd Asia-Pacific Software Engineering Conference (APSEC)*, 2016, pp. 153–160. doi:10.1109/APSEC.2016.031

[80] J. Scheuner and P. Leitner, "Replication package for "function-as-a-service performance evaluation: a multivocal literature review"," v1.0, 2020, dataset. doi:10.5281/zenodo.3906613

[81] V. Garousi, M. Felderer, and T. Hacaloglu, "Software test maturity assessment and test process improvement: A multivocal literature review," *Information and Software Technology*, pp. 16–42, 2017. doi:10.1016/j.infsof.2017.01.001

[82] L. Wang, M. Li, Y. Zhang, T. Ristenpart, and M. Swift, "Peeking behind the curtains of serverless platforms," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2018, pp. 133–146. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/wang-liang

[83] P. Leitner and J. Scheuner, "Bursting with possibilities – an empirical study of credit-based bursting cloud instance types," in *Proceedings of the 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2015, pp. 227–36. doi:10.1109/UCC.2015.39

[84] C. Davatz, C. Inzinger, J. Scheuner, and P. Leitner, "An approach and case study of cloud instance type selection for multi-tier web applications," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 534–543. doi:10.1109/CCGRID.2017.12

[85] C. Boettiger, "An introduction to docker for reproducible research," *Operating Systems Review*, pp. 71–79, 2015. doi:10.1145/2723872.2723882

[86] S. Eismann, J. Scheuner, E. van Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L. Abad, and A. Iosup, "SPEC RG technical report: A review of serverless use cases and their characteristics – dataset," 2020. doi:10.5281/zenodo.3822191

[87] E. V. Eyk, A. Iosup, J. Grohmann, S. Eismann, A. Bauer, L. Versluis, L. Toader, N. Schmitt, N. Herbst, and C. L. Abad, "The SPEC-RG reference architecture for FaaS: From microservices and containers to serverless platforms," *IEEE Internet Computing*, pp. 7–18, 2019. doi:10.1109/MIC.2019.2952061

[88] V. Yussupov, U. Breitenbücher, A. Kaplan, and F. Leymann, "Seaport: Assessing the portability of serverless applications," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER)*, 2020, pp. 456–467. doi:10.5220/0009574104560467

[89] E. V. Eyk, J. Scheuner, S. Eismann, C. L. Abad, and A. Iosup, "Beyond microbenchmarks: The SPEC-RG vision for a comprehensive serverless benchmark," in *Companion of the 11th ACM/SPEC ICPE: 3rd Workshop on Hot Topics in Cloud Computing Performance (HotCloudPerf)*, 2020, pp. 26–31. doi:10.1145/3375555.3384381

[90] B. Farley, A. Juels, V. Varadarajan, T. Ristenpart, K. D. Bowers, and M. M. Swift, "More for your money: exploiting performance heterogeneity in public clouds," in *Proceedings of the 3rd ACM Symposium on Cloud Computing (SoCC)*, 2012. doi:10.1145/2391229.2391249

[91] T. Palit, Y. Shen, and M. Ferdman, "Demystifying cloud benchmarking," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2016, pp. 122–132. doi:10.1109/ISPASS.2016.7482080

[92] J. Dejun, G. Pierre, and C.-H. Chi, "EC2 performance analysis for resource provisioning of service-oriented applications," in *Proceedings of the 7th ICSOC / 2nd ServiceWave Workshops*, 2009, pp. 197–207. doi:10.1007/978-3-642-16132-2_19

[93] J. Scheuner, J. Cito, P. Leitner, and H. Gall, "Cloud WorkBench: Benchmarking IaaS providers based on infrastructure-as-code," in *Companion of the 24th International Conference on World Wide Web (WWW Demo)*, 2015, pp. 239–242. doi:10.1145/2740908.2742833

[94] C. Spectator, "2017 top 10 european cloud providers," Cloud Spectator, Tech. Rep., 2017. [Online]. Available: https://cloudspectator.com/top-10-european-cloud-service-providers/

[95] (2016) SPEC Cloud™ IaaS 2016 Benchmark. [Online]. Available: http://spec.org/cloud_iaas2016/

[96] M. Cunha, N. Mendonça, and A. Sampaio, "A declarative environment for automatic performance evaluation in IaaS clouds," in *Proceedings of the 6th IEEE International Conference on Cloud Computing (CLOUD)*, 2013, pp. 285–92. doi:10.1109/CLOUD.2013.12

[97] J. Scheuner, "Cloud benchmarking – estimating cloud application performance based on micro benchmark profiling," Master Thesis, University of Zurich, 2017. [Online]. Available: https://www.merlin.uzh.ch/publication/show/15364

[98] B. Gregg, *Systems Performance: Enterprise and the Cloud*. Prentice Hall, 2013. [Online]. Available: http://books.google.ch/books?id=pTYkAQAAQBAJ

[99] J. Scheuner and P. Leitner, "A cloud benchmark suite combining micro and applications benchmarks," in *Companion of the 9th ACM/SPEC ICPE: 4th Workshop on Quality-Aware DevOps (QUDOS)*, 2018, pp. 161–166. doi:10.1145/3185768.3186286

[100] J. O'Loughlin and L. Gillam, "Towards performance prediction for public infrastructure clouds: An EC2 case study," in *Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2013, pp. 475–80. doi:10.1109/CloudCom.2013.69

[101] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2010, pp. 1–14. doi:10.1145/1879141.1879143

[102] D. Cerotti, M. Gribaudo, P. Piazzolla, and G. Serazzi, "Flexible CPU provisioning in clouds: A new source of performance unpredictability," in *Proceedings of the 9th International Conference on Quantitative Evaluation of Systems (QEST)*, 2012, pp. 230–37. doi:10.1109/QEST.2012.23

[103] S. K. Barker and P. Shenoy, "Empirical evaluation of latency-sensitive application performance in the cloud," in *Proceedings of the 1st ACM SIGMM Conference on Multimedia Systems (MMSys)*, 2010, pp. 35–46. doi:10.1145/1730836.1730842

[104] M. Canuto, R. Bosch, M. Macias, and J. Guitart, "A methodology for full-system power modeling in heterogeneous data centers," in *Proceedings of the 9th International Conference on Utility and Cloud Computing (UCC)*, 2016, pp. 20–29. doi:10.1145/2996890.2996899

[105] K. Hoste, A. Phansalkar, L. Eeckhout, A. Georges, L. K. John, and K. De Bosschere, "Performance prediction based on inherent program similarity," in *Proceedings of the 15th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2006, pp. 114–122. doi:10.1145/1152154.1152174

[106] C. Stewart and K. Shen, "Performance modeling and system management for multi-component online services," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design*

& Implementation (NSDI), 2005, pp. 71–84. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251203.1251209

[107] Y. El-Khamra, H. Kim, S. Jha, and M. Parashar, "Exploring the performance fluctuations of HPC workloads on clouds," in *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 383–87. doi:10.1109/CloudCom.2010.84

[108] B. G. Tabachnick, L. S. Fidell, and J. B. Ullman, *Using Multivariate Statistics*. Pearson, 2012.

[109] C. Lowery, R. Bala, L. Leong, and D. Smith, "Magic quadrant for cloud infrastructure as a service, worldwide," Gartner Research, Tech. Rep., 2017. [Online]. Available: https://www.gartner.com/en/documents/3738058/magic-quadrant-for-cloud-infrastructure-as-a-service-wor

[110] (2017) $7.72 billion function-as-a-service market 2017. [Online]. Available: https://bwnews.pr/2VBDBgC

[111] J. M. Hellerstein, J. M. Faleiro, J. Gonzalez, J. Schleier-Smith, V. Sreekanti, A. Tumanov, and C. Wu, "Serverless computing: One step forward, two steps back," in *Proceedings of the 9th Conference on Innovative Data Systems Research (CIDR)*, 2019. [Online]. Available: http://cidrdb.org/cidr2019/papers/p119-hellerstein-cidr19.pdf

[112] (2020) Accelerating with serverless! [Online]. Available: https://medium.com/lego-engineering/accelerating-with-serverless-625da076964b

[113] A. Eivy, "Be wary of the economics of "serverless" cloud computing," *IEEE Cloud Computing*, pp. 6–2, 2017. doi:10.1109/MCC.2017.32

[114] J. Demian. (2018) Serverless case study - netflix. [Online]. Available: https://dashbird.io/blog/serverless-case-study-netflix/

[115] (2019) Worldwide IaaS public cloud services market grew 31.3[Online]. Available: https://bwnews.pr/2ZcI7o4

[116] N. Malishev. (2019) AWS Lambda cold start language comparisons, 2019 edition. [Online]. Available: https://levelup.gitconnected.com/aws-lambda-cold-start-language-comparisons-2019-edition-%EF%B8%8F-1946d32a0244

[117] I. Baldini, P. Cheng, S. J. Fink, N. Mitchell, V. Muthusamy, R. Rabbah, P. Suter, and O. Tardieu, "The serverless trilemma: Function composition for serverless computing," in *Proceedings of the ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, 2017, pp. 89–103. doi:10.1145/3133850.3133855

[118] J. Spillner, C. Mateos, and D. A. Monge, "Faaster, better, cheaper: The prospect of serverless scientific computing and HPC," in *Proceedings of the 4th Latin American Conference on High Performance Computing (CARLA)*, 2017, pp. 154–168. doi:10.1007/978-3-319-73353-1_11

[119] V. Ishakian, V. Muthusamy, and A. Slominski, "Serving deep learning models in a serverless platform," in *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E)*, 2018, pp. 257–62. doi:10.1109/IC2E.2018.00052

[120] J. Manner, M. Endreß, T. Heckel, and G. Wirtz, "Cold start influencing factors in function as a service," in *Companion of the 11th IEEE/ACM UCC: 4th International Workshop on Serverless Computing (WoSC)*, 2018, pp. 181–88. doi:10.1109/UCC-Companion.2018.00054

[121] K. Figiela, A. Gajek, A. Zima, B. Obrok, and M. Malawski, "Performance evaluation of heterogeneous cloud functions," *Concurrency and Computation: Practice and Experience*, 2018. doi:10.1002/cpe.4792

[122] I. Pelle, J. Czentye, J. Dóka, and B. Sonkoly, "Towards latency sensitive cloud native applications: A performance study on AWS," in *Proceedings of the 12th IEEE International Conference on Cloud Computing (CLOUD)*, 2019, pp. 272–280. doi:10.1109/CLOUD.2019.00054

[123] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, and B. Regnell, *Experimentation in Software Engineering.* Springer, 2012. doi:10.1007/978-3-642-29044-2

[124] M. Shahrad, J. Balkind, and D. Wentzlaff, "Architectural implications of function-as-a-service computing," in *Proceedings of the 52nd IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2019, pp. 1063–1075. doi:10.1145/3352460.3358296

[125] J. Scheuner and P. Leitner, "Estimating cloud application performance based on micro-benchmark profiling," in *Proceedings of the 11th IEEE International Conference on Cloud Computing (CLOUD)*, 2018, pp. 90–97. doi:10.1109/CLOUD.2018.00019

[126] J. Kuhlenkamp, S. Werner, M. C. Borges, D. Ernst, and D. Wenzel, "Benchmarking elasticity of FaaS platforms as a foundation for objective-driven design of serverless applications," in *Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing (SAC)*, 2020, pp. 1576–1585. doi:10.1145/3341105.3373948

[127] J. Kuhlenkamp and M. Klems, "Costradamus: A cost-tracing system for cloud-based software services," in *Proceedings of the 15th International Conference on Service-Oriented Computing (ICSOC)*, 2017, pp. 657–672. doi:10.1007/978-3-319-69035-3_48

[128] P. J. Fleming and J. J. Wallace, "How not to lie with statistics: The correct way to summarize benchmark results," *Communications of the ACM*, pp. 218–221, 1986. doi:10.1145/5666.5673

[129] M. Al-Ameen and J. Spillner, "Systematic and open exploration of FaaS and serverless computing research," in *Proceedings of the European Symposium on Serverless Computing and Applications (ESSCA)*, 2018, pp. 30–35. [Online]. Available: http://ceur-ws.org/Vol-2330/short2.pdf

[130] Z. Li, L. O'Brien, R. Cai, and H. Zhang, "Towards a taxonomy of performance evaluation of commercial cloud services," in *Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD)*, 2012, pp. 344–51. doi:10.1109/CLOUD.2012.74

[131] N. Bjørndal, A. Bucchiarone, M. Mazzara, N. Dragoni, S. Dustdar, F. B. Kessler, and T. Wien, "Migration from monolith to microservices: Benchmarking a case study," 2020, unpublished. doi:10.13140/RG.2.2.27715.14883

[132] I. E. Akkus, R. Chen, I. Rimac, M. Stein, K. Satzke, A. Beck, P. Aditya, and V. Hilt, "SAND: towards high-performance serverless computing," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2018, pp. 923–935. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/akkus

[133] L. F. Albuquerque Jr, F. S. Ferraz, R. F. Oliveira, and S. M. Galdino, "Function-as-a-service x platform-as-a-service: Towards a comparative study on FaaS and PaaS," *ICSEA 2017*, p. 217, 2017. [Online]. Available: https://www.thinkmind.org/download.php?articleid=icsea_2017_9_30_10096

[134] T. Back and V. Andrikopoulos, "Using a microbenchmark to compare function as a service solutions," in *Proceedings of the 7th European Service-Oriented and Cloud Computing (ESOCC)*, 2018, pp. 146–160. doi:10.1007/978-3-319-99819-0_11

[135] D. Balla, M. Maliosz, C. Simon, and D. Gehberger, "Tuning runtimes in open source FaaS," in *Proceedings of the Internet of Vehicles. Technologies and Services Toward Smart Cities (IOV)*, 2020, pp. 250–266. doi:10.1007/978-3-030-38651-1_21

[136] D. Bardsley, L. Ryan, and J. Howard, "Serverless performance and optimization strategies," in *Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud)*, 2018, pp. 19–6. doi:10.1109/SmartCloud.2018.00012

[137] D. Bortolini and R. R. Obelheiro, "Investigating performance and cost in function-as-a-service platforms," in *Proceedings of the 14th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2019, pp. 174–185. doi:10.1007/978-3-030-33509-0_16

[138] J. Carreira, P. Fonseca, A. Tumanov, A. Zhang, and R. Katz, "A case for serverless machine learning," in *Workshop on Systems for ML and Open Source Software at NeurIPS*, 2018. [Online]. Available: http://learningsys.org/nips18/assets/papers/101CameraReadySubmissioncirrus_nips_final2.pdf

[139] Y. Gan, Y. Zhang, D. Cheng, A. Shetty, P. Rathi, N. Katarki, A. Bruno, J. Hu, B. Ritchken, B. Jackson, K. Hu, M. Pancholi, Y. He, B. Clancy, C. Colen, F. Wen, C. Leung, S. Wang, L. Zaruvinsky, M. Espinosa, R. Lin, Z. Liu, J. Padilla, and C. Delimitrou, "An open-source benchmark suite

for microservices and their hardware-software implications for cloud & edge systems," in *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019, pp. 3–18. doi:10.1145/3297858.3304013

[140] V. Giménez-Alventosa, G. Moltó, and M. Caballer, "A framework and a performance assessment for serverless mapreduce on AWS Lambda," *Future Generation Computer Systems*, pp. 259–274, 2019. doi:10.1016/j.future.2019.02.057

[141] V. Gupta, S. Wang, T. A. Courtade, and K. Ramchandran, "Oversketch: Approximate matrix multiplication for the cloud," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, 2018, pp. 298–304. doi:10.1109/BigData.2018.8622139

[142] A. Hall and U. Ramachandran, "An execution model for serverless functions at the edge," in *Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI)*, 2019, pp. 225–236. doi:10.1145/3302505.3310084

[143] C. Ivan, R. Vasile, and V. Dadarlat, "Serverless computing: An investigation of deployment environments for Web APIs," *Computers*, 2019. doi:10.3390/computers8020050

[144] D. Jackson and G. Clynch, "An investigation of the impact of language runtime on the performance and cost of serverless functions," in *Companion of the 11th IEEE/ACM UCC: 4th International Workshop on Serverless Computing (WoSC)*, 2018, pp. 154–60. doi:10.1109/UCC-Companion.2018.00050

[145] Q. Jiang, Y. C. Lee, and A. Y. Zomaya, "Serverless execution of scientific workflows," in *Proceedings of the 15th International Conference on Service-Oriented Computing (ICSOC)*, 2017, pp. 706–721. doi:10.1007/978-3-319-69035-3_51

[146] J. Kim and K. Lee, "FunctionBench: A suite of workloads for serverless cloud function service," in *Proceedings of the 12th IEEE International Conference on Cloud Computing (CLOUD WIP)*, 2019, pp. 502–504. doi:10.1109/CLOUD.2019.00091

[147] J. Kim, J. Park, and K. Lee, "Network resource isolation in serverless cloud function service," in *Proceedings of the 7th International Workshop on Autonomic Management of High-Performance Grid and Cloud Computing (AMGCC) at ICAC/SASO*, 2019, pp. 182–187. doi:10.1109/FAS-W.2019.00051

[148] A. Klimovic, Y. Wang, C. Kozyrakis, P. Stuedi, J. Pfefferle, and A. Trivedi, "Understanding ephemeral storage for serverless analytics," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2018, pp. 789–794. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/klimovic-serverless

[149] J. Kuhlenkamp, S. Werner, M. C. Borges, K. El Tal, and S. Tai, "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)*, 2019, pp. 1–9. doi:10.1145/3344341.3368796

[150] H. Lee, K. Satyam, and G. C. Fox, "Evaluation of production serverless computing environments," in *Proceedings of the 11th IEEE CLOUD: 3rd International Workshop on Serverless Computing (WoSC)*, 2018, pp. 442–50. doi:10.1109/CLOUD.2018.00062

[151] J. Li, S. G. Kulkarni, K. K. Ramakrishnan, and D. Li, "Understanding open source serverless platforms: Design considerations and performance," in *Proceedings of the 5th International Workshop on Serverless Computing (WoSC) at Middleware*, 2019, pp. 37–42. doi:10.1145/3366623.3368139

[152] W. Lloyd, S. Ramesh, S. Chinthalapati, L. Ly, and S. Pallickara, "Serverless computing: An investigation of factors influencing microservice performance," *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E)*, 2018. doi:10.1109/IC2E.2018.00039

[153] W. Lloyd, M. Vu, B. Zhang, O. David, and G. Leavesley, "Improving application migration to serverless computing platforms: Latency mitigation with keep-alive workloads," in *Companion of the 11th IEEE/ACM UCC: 4th International Workshop on Serverless Computing (WoSC)*, 2018, pp. 195–00. doi:10.1109/UCC-Companion.2018.00056

[154] P. G. López, M. Sánchez-Artigas, G. París, D. B. Pons, Á. R. Ollobarren, and D. A. Pinto, "Comparison of FaaS orchestration systems," in *Companion of the 11th IEEE/ACM UCC: 4th International Workshop on Serverless Computing (WoSC)*, 2018, pp. 148–53. doi:10.1109/UCC-Companion.2018.00049

[155] M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, "Serverless execution of scientific workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions," *Future Generation Computer Systems*, 2017. doi:10.1016/j.future.2017.10.029

[156] S. Malla and K. Christensen, "HPC in the cloud: Performance comparison of function as a service (FaaS) vs infrastructure as a service (IaaS)," *Internet Technology Letters*, 2019. doi:10.1002/itl2.137

[157] G. McGrath and P. R. Brenner, "Serverless computing: Design, implementation, and performance," in *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2017, pp. 405–10. doi:10.1109/ICDCSW.2017.36

[158] A. Mohan, H. Sane, K. Doshi, S. Edupuganti, N. Nayak, and V. Sukhomlinov, "Agile cold starts for scalable serverless," in *Proceedings of the 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*, 2019. [Online]. Available: https://www.usenix.org/conference/hotcloud19/presentation/mohan

[159] S. K. Mohanty, G. Premsankar, and M. D. Francesco, "An evaluation of open source serverless computing frameworks," in *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2018, pp. 115–120. doi:10.1109/CloudCom2018.2018.00033

[160] X. Niu, D. Kumanov, L. Hung, W. Lloyd, and K. Y. Yeung, "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)*, 2019, pp. 683–687. doi:10.1145/3307339.3343465

[161] E. Oakes, L. Yang, D. Zhou, K. Houck, T. Harter, A. Arpaci-Dusseau, and R. Arpaci-Dusseau, "SOCK: Rapid task provisioning with serverless-optimized containers," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2018, pp. 57–70. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/oakes

[162] A. Pérez, G. Moltó, M. Caballer, and A. Calatrava, "A programming model and middleware for high throughput serverless computing applications," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC)*, 2019, pp. 106–113. doi:10.1145/3297280.3297292

[163] Q. Pu, S. Venkataraman, and I. Stoica, "Shuffling, fast and slow: Scalable analytics on serverless infrastructure," in *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2019, pp. 193–206. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/pu

[164] H. Puripunpinyo and M. H. Samadzadeh, "Effect of optimizing Java deployment artifacts on AWS Lambda," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017, pp. 438–43. doi:10.1109/INFCOMW.2017.8116416

[165] A. Saha and S. Jindal, "EMARS: efficient management and allocation of resources in serverless," in *Proceedings of the 11th IEEE International Conference on Cloud Computing (CLOUD)*, 2018, pp. 827–830. doi:10.1109/CLOUD.2018.00113

[166] S. Shillaker, "A provider-friendly serverless framework for latency-critical applications," in *12th Eurosys Doctoral Workshop*, 2018. [Online]. Available: http://conferences.inf.ed.ac.uk/EuroDW2018/papers/eurodw18-Shillaker.pdf

[167] A. Singhvi, S. Banerjee, Y. Harchol, A. Akella, M. Peek, and P. Rydin, "Granular computing and network intensive applications: Friends or foes?" in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, 2017, pp. 157–163. doi:10.1145/3152434.3152450

[168] S. Werner, J. Kuhlenkamp, M. Klems, J. Müller, and S. Tai, "Serverless big data processing using matrix multiplication as example," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, 2018, pp. 358–65. doi:10.1109/BigData.2018.8622362

[169] M. Zhang, Y. Zhu, C. Zhang, and J. Liu, "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*, 2019, pp. 61–66. doi:10.1145/3304112.3325608

[170] K. Kimura, A. Sekiguchi, S. Choudhary, and T. Uehara, "A JavaScript transpiler for escaping from complicated usage of cloud services and APIs," in *Proceedings of the 25th Asia-Pacific Software Engineering Conference (APSEC)*, 2018, pp. 69–78. doi:10.1109/APSEC.2018.00021

[171] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, "Network requirements for resource disaggregation," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 249–264. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/gao