COLLEGE OF ENGINEERING
MASTER OF SCIENCE IN COMPUTER SCIENCE

Master's Thesis Proposal

Student Name: _____Jeremiah Webb_____Student ID #__2545328

Tentative Thesis Title: __Comparative Performance Analysis of Cryptographic Workloads Across Cloud Providers: A Multi-Language Study on FaaS and IaaS Platforms_____
Anticipated Graduation Date/Term: ___Spring 2025_____

Signed by the Committee: Written Proposal has been passed

Student: Jeremiah Webb — Dept: EECS

Signed: Jeremiah Webb — Jeremiah Webb (Sep 6, 2024 15:15 EDT) — Date: 09/06/2024

Advisor/Chair: Dr. Laxima Niure Kandel — Dept: EECS

Signed: Dr. Laxima Niure Kandel — Dr. Laxima Niure Kandel (Sep 6, 2024 15:19 EDT) — Date: 09/06/2024

Co-Advisor: Dr. Omar Ochoa — Dept: EECS

Signed: — Date: 09/09/2024

OR:

Member: Dr. David Bethelmy — Dept: EECS

Signed: Dr. David Bethelmy (Sep 7, 2024 19:32 EDT) — Date: 09/07/2024

Member: Dr. Shafika Showkat Moni — Dept: EECS

Signed: Dr. Shafika Showkat Moni (Sep 6, 2024 17:08 EDT) — Date: 09/06/2024

Following acceptance of the written proposal by the committee, return this form to the Graduate Program Coordinator. ***This form must be received prior to any term in which the student is enrolled for Master's Thesis hours.***

Program Coordinator: — Date: 09/09/2024

Form Identification: EECE-Thesis_proposal_form_2021 (Thesis Proposal Approval Page)

Comparative Performance Analysis of Cryptographic Workloads Across Cloud Providers: A Multi-Language Study on FaaS and IaaS Platforms

A thesis proposal submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science at Embry-Riddle Aeronautical University 2024

Abstract
This study will provide a performance analysis of common cryptographic workloads across Function as a Service (FaaS) and Infrastructure as a Service (IaaS) offerings from AWS and Azure using microbenchmarks. By utilizing the Amazon Web Services (AWS) Cloud Development Kit (CDK) and Azure Resource Manager (ARM) templates, the study will demonstrate how to systematically evaluate cryptographic workloads on AWS Lambda functions, Elastic Compute Cloud (EC2) instances, and their Azure equivalents as Infrastructure-as-code (IaC). The result of the study will be tables of performance metrics based on diverse architectural, software and hardware configurations such as, x86 and Arm architectures, various programming languages (Rust, Go, Python, Java, C#, TypeScript), and memory sizes ranging from 128MB to 10GB.

Keywords: cloud computing, cryptography, microbenchmark, performance analysis, infrastructure-as-code

Introduction

Cloud computing has revolutionized how we access and manage computing resources. Currently, cloud computing stands 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 software services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Cloud computing has revolutionized how we access and manage computing resources. The flexibility and scalability of cloud computing have made it indispensable for businesses and individuals.

Among the leading cloud platforms today are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform, each offering a diverse array of services designed to meet various computing needs. These platforms offer multiple service models, including Infrastructure-as-a-service (IaaS), Function-as-a-service (FaaS), and Software-as-a-Service (SaaS). These models allow users to choose the level of control and management they require, from fully managed services to more customizable infrastructure options [2]. Cloud computing forms a dynamic IT resource pool where users can upload their data to the cloud for storage and computation, benefiting from the cloud's ability to optimize resource usage and reduce overhead costs [3]. Using cloud resources therefore may offer less complex storage solutions and a reduction in maintenance costs [4].

However, this convenience comes with a trade-off: once data is uploaded to the cloud, users relinquish physical control over it. This loss of control has raised significant concerns about trust, data privacy, and the security of sensitive information [3]. To mitigate cybersecurity concerns, cloud users are increasingly turning to various forms of encryption as a solution to protect their data before storing it in the cloud [5]. By

encrypting data, users can safeguard their privacy when using shared computing resources alongside other customers [9]. Cryptographic algorithms can be computationally expensive. By studying performance metrics, one can determine which cryptographic algorithms to use and under which conditions.

Background

Cloud Computing Paradigms

This study will consider two popular cloud computing paradigms: Function-as-a-Service (FaaS) and Infrastructure-as-a-Service (IaaS).

FaaS is a serverless cloud computing paradigm and is defined through FaaS platforms, such as AWS Lambda and Azure Functions [6]. These services execute code snippets written in various programming languages such as Python, Node.js and Go. These services can be triggered by events such as incoming HTTP requests [6]. FaaS is used for a large variety of tasks such as the implementation of event-based data ingestion pipelines for machine learning [7], the provision of RESTful Application Programming Interfaces (API), the analysis of data of network performance [8], and the facilitation encryption of data [9]. In addition, it can be used to ‘glue’ together larger serverless applications.

IaaS provides virtual IT resources such as compute, storage, and networking platforms. These resources are offered as on-demand services by the cloud service providers (CSPs) [10], namely, AWS and Azure. Services like AWS EC2 and Azure Virtual Machines provide various options that allow the scaling of RAM, virtual CPUs (vCPUs), network bandwidth, and storage performance. These options allow for optimal capacity planning and system management to meet the performance requirements of specific workloads [11].

Infrastructure-as-Code

Infrastructure-as-Code (IaC) is the practice of automatically defining and managing deployment environments, system configurations, and cloud infrastructure through source code. IaC involves writing and maintaining infrastructure definitions in code, which can be versioned, reviewed, and tested just like application code [12]. The "as code" suffix signifies the adoption of software development best practices, such as version control for scripts, automated testing, and incremental code changes [13]. IaC is susceptible to programming mistakes and can cause disruptions and outages [13].

Microbenchmarking

Microbenchmarking refers to obtaining usage and efficiency metrics through a variety of tools provided by CSPs. A microbenchmark is a performance test that runs short tasks (e.g., less than 1ms) on small software units, such as individual methods. These tests are conducted in isolation, without requiring a fully deployed system. Microbenchmarks are

executed repeatedly over a set period (e.g., 1 second) to measure the average execution time. The results are reported as the distribution of multiple iterations (e.g., 20) [14]. In particular, software microbenchmarks can be seen as the unit-test equivalent of performance for a microservice[1] [15].

Related Work

There have been many benchmarks relating to IaaS and FaaS paradigms amongst cloud providers. For example, in 2018, researchers at Indiana University Bloomington attempted to measure performance values for CPU usage, max function throughput, and programming language run time amongst AWS Lambda, Google Functions and Azure functions [17]. They found that these FaaS offerings were more cost-effective than their IaaS counterparts (AWS EC2 and Azure Virtual Machine) due to the almost zero delay bootup[2]. This research did not use cryptographic workloads to do their testing but focused on multiplying matrices and uploading/downloading files to determine I/O speed and response times for concurrent invocations of a function.

In 2023, Cascadeo did an analysis to determine performance and cost savings between x86 and Arm architectures on AWS Lambda (FaaS) [18]. By using IaC, Cascadeo focused on measuring speed and RAM used by provisioning 1,680 unique Lambda functions (2 architectures x 7 runtimes x 5 workloads x [1-6] workers x [6-13] possible memory configs). The experiment discussed multiple workloads including high memory workloads, I/O operation workloads, and single and multi-threaded workloads which represent common use cases in AWS Lambda. However, this experiment did not go into common cryptographic workloads; it only conducted SHA-256 cryptographic hashing to simulate CPU load. Unfortunately, Cascadeo's research currently does not have public code or raw data from the experiment.

Problem Identification

Previous research conducted microbenchmarks measuring the performance of programming languages on FaaS and IaaS platforms. However, these microbenchmarks relied on simulated workloads. However, simulated workloads fail to capture realistic performance metrics, particularly for cryptographic operations. Addressing performance issues early is crucial for preventing poor user experiences, excessive resource consumption, and significant maintenance efforts, all of which can lead to unpredictable additional costs [19]. Previous work does not provide sufficient performance metrics relating to cryptographic workloads in cloud environments. This study aims to fill that gap. To that end, the following algorithms have been chosen: AES-256-GCM, RSA-2048, RSA-3072, RSA-4096, SHA-256, SHA-384, ECC NIST P-256, ECC NIST P-384 [20]

---

[1] Microservices are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs. [16]

[2] Delay bootup refers to the time required for the CSP to provision the selected service.

[21]. These algorithms were chosen because they are supported by AWS and Azure. To determine these performance metrics for cloud environments, the study will focus on the following 2 questions:

RQ1. Will Azure and AWS performance differ for identical configurations in IaaS for each algorithm?

RQ2. Will Azure and AWS performance differ for identical configurations in FaaS for each algorithm?

Methodology

This study will attempt to retrieve performance metric data of cryptographic workloads by leveraging the IaC platforms AWS Cloud Development Kit (CDK) and Azure Resource Management Templates (ARM). The IaC will then be used to deploy and manage the cloud resources, ensuring consistency and repeatability across different configurations. This method will enable a thorough analysis of performance metrics by running microbenchmarks on these configurations. The approach will involve the systematic development of cryptographic workloads tailored for cloud infrastructure, specifically within AWS and Azure IaaS and FaaS paradigms. These cryptographic workloads will be implemented in Rust, Go, Python, Java, C#, and TypeScript. The study will follow best practices identified through comprehensive research of provided documentation of AWS and Azure. See Figure 1 for an overall view of the microbenchmarking process.

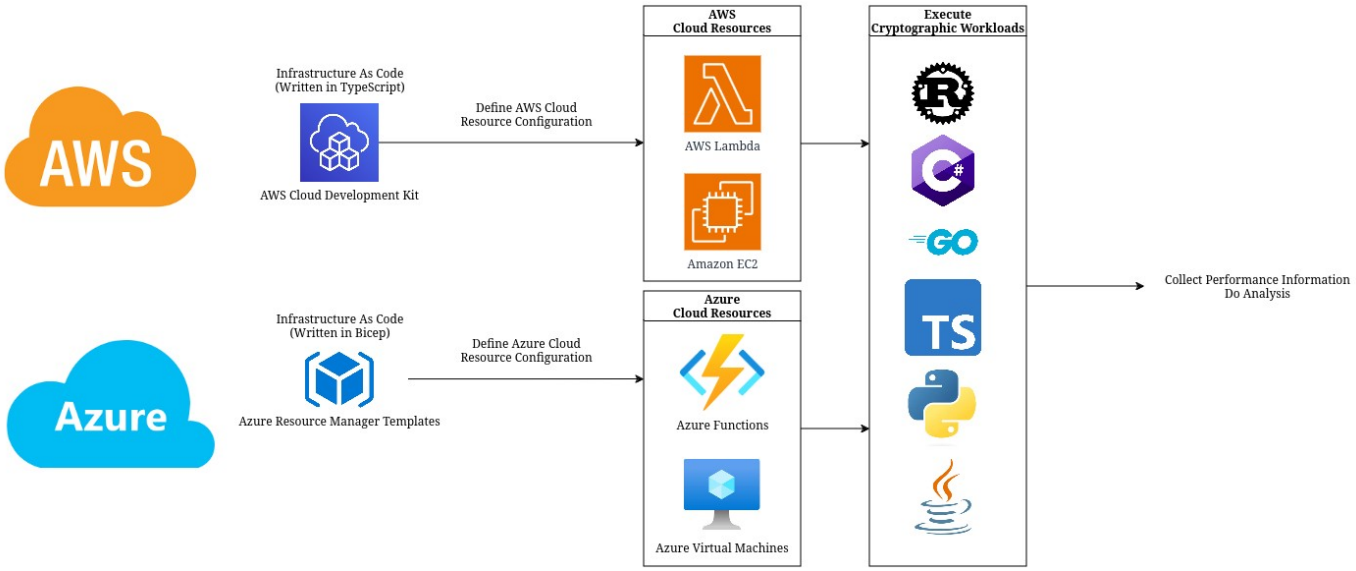

Figure 1, The proposed AWS and Azure microbenchmarking process

The study will utilize each CSP's IaC platform to create the needed cloud resources asynchronously. The AWS Cloud Development Kit (written in TypeScript) will define resources such as AWS Lambda and Amazon EC2, while Azure will define Resource Manager Templates (written in Bicep) to specify Azure Functions and Virtual Machines. These cloud resources will be configured to run the cryptographic algorithms previously mentioned and implemented in the programming languages Rust, C#, Go, TypeScript, Python, and Java. The performance data collected from these individual workloads will then be analyzed to evaluate efficiency and resource utilization across different configurations. A microbenchmark for the resources will include the ability to be cold started and warmed up.

Project Deliverables and Timeline

The study will be completed over two consecutive semesters, the Fall 2024 and the Spring 2025 semesters. The overall delivery schedule can be seen in *Table 1*.

During the Fall 2024 semester, the study will include a comprehensive literature review on implementing cryptographic workloads on cloud providers, with a focus on Azure and AWS IaaS and FaaS paradigms. This review will include Azure and AWS documentation, industry white papers, and open-source documentation that benchmarks workloads on cloud resources. Within the same timeframe, the study will also identify and establish best practices for writing cryptographic workloads in Rust, Go, Python, Java, C#, and TypeScript. By the end of the Fall 2024 semester, this researcher will implement each cryptographic algorithm using each programming language.

During the Spring 2025 semester, the focus will be on finalizing and writing the code for the IaC for AWS and Azure. This semester will also involve collecting and analyzing the performance data from each cloud provider. These results will be documented in a written final report and orally defended. The study's code will be stored in a GitHub repository.

Table 1: Project deliverables and timeline

#	Deliverable	Completion Date
1	Proposal and initial literature review	08/26/2024
2	Continued in-depth literature review	9/26/2024
3	Requirements and system design definitions of FaaS and IaaS cryptographic workloads on AWS and Azure	11/18/2024
5	Develop programs for cryptographic workloads on AWS and Azure.	12/4/2024

5	Create infrastructure as code for AWS	1/31/2025
6	Create infrastructure as code for Azure	1/31/2025
7	AWS testing and result analysis	2/28/2025
8	Azure testing and result analysis	3/28/2025
9	Final report writing and editing	4/21/2025
10	Thesis defense	TBD

References

[1] Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology.

[2] Growing At a CAGR of 17.3 % - The Public Cloud Market Size Projected to Grow $987.7 billion by 2027, Report by MarketsandMarket (2023, Feb 23). *NASDAQ OMX's News Release Distribution Channel* https://www.proquest.com/wire-feeds/growing-at-cagr-17-3-public-cloud-market-size/docview/2778890789/se-2

[3] Xiaoyu, W., & Zhengming, G. (2020). Research and Development of Data Security Multidimensional Protection System in Cloud Computing Environment. *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*, 67–70. https://doi.org/10.1109/ICAACI50733.2020.00019

[4] Lan Zhou, Varadharajan, V., & Hitchens, M. (2015). Trust Enhanced Cryptographic Role-Based Access Control for Secure Cloud Data Storage. *IEEE Transactions on Information Forensics and Security*, *10*(11), 2381–2395. https://doi.org/10.1109/TIFS.2015.2455952

[5] Mohammed, S., Nanthini, S., Bala Krishna, N., Srinivas, I. V., Rajagopal, M., & Ashok Kumar, M. (2023). A new lightweight data security system for data security in the cloud computing. *Measurement. Sensors*, *29*, 100856-. https://doi.org/10.1016/j.measen.2023.100856

[6] Scheuner, J., & Leitner, P. (2020). Function-as-a-Service performance evaluation: A multivocal literature review. *The Journal of Systems and Software*, *170*, 110708-. https://doi.org/10.1016/j.jss.2020.110708

[7] Mathew, A., Andrikopoulos, V., & Blaauw, F. J. (2021). Exploring the cost and performance benefits of AWS step functions using a data processing pipeline. *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*, 1–10. https://doi.org/10.1145/3468737.3494084

[8] Jonas, E., Pu, Q., Venkataraman, S., Stoica, I., & Recht, B. (2017). *Occupy the Cloud: Distributed Computing for the 99%*

[9] Subramanian, E. K., & Tamilselvan, L. (2020). Elliptic curve Diffie–Hellman cryptosystem in big data cloud security. *Cluster Computing*, *23*(4), 3057–3067. https://doi.org/10.1007/s10586-020-03069-3

[10] Zhang, Y., Krishnan, R., & Sandhu, R. (2014). Secure Information and Resource Sharing in Cloud Infrastructure as a Service. *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, 81–90. https://doi.org/10.1145/2663876.2663884

[11] Jiang Y, Adams B (2015) Co-evolution of infrastructure and source code: an empirical study. In: Proceedings of the 12th working conference on mining software repositories, ieee press, Piscataway, NJ, USA, MSR'15, pp 45–55. http://dl.acm.org/citation.cfm?id=2820518.282052

[12] Humble J, Farley D (2010) Continuous delivery: reliable software releases through build, test, and deployment automation, 1st. Addison-Wesley Professional, Boston

[13] Morris K (2016) Infrastructure as code: managing servers in the cloud. O'Reilly Media, Inc.

[14] Leitner P, Cito J (2016) Patterns in the chaos–a study of performance variation and predictability in public iaas clouds. ACM Trans Internet Technol 16(3):15:1–15:23. https://doi.org/10.1145/2885497

[15] Laaber, C., Scheuner, J., & Leitner, P. (2019). Software microbenchmarking in the cloud. How bad is it really? *Empirical Software Engineering, 24*(4), 2469-2508. https://doi.org/10.1007/s10664-019-09681-1

[16] Amazon Web Services (2024) What are Microservices? https://aws.amazon.com/microservices/

[17] Lee, H., Satyam, K., & Fox, G. (2018). Evaluation of Production Serverless Computing Environments. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 442–450. https://doi.org/10.1109/CLOUD.2018.00062

[18] Roig, J., Choudhury, D., DeMuth, J., & Singla, R. (2023, October 14). *Comparing aws lambda arm vs. x86 performance, cost, and analysis | AWS Partner Network (APN) blog*. Amazon Web Services. https://aws.amazon.com/blogs/apn/comparing-aws-lambda-arm-vs-x86-performance-cost-and-analysis-2/

[19] Grambow, M., Kovalev, D., Laaber, C., Leitner, P., & Bermbach, D. (2023). Using Microbenchmark Suites to Detect Application Performance Changes. *IEEE Transactions on Cloud Computing*, *11*(3), 2575–2590. https://doi.org/10.1109/TCC.2022.3217947

[20] Amazon Web Services. (2024). *FAQs | AWS Key Management Service (KMS) | Amazon Web Services (AWS)*. Amazon Web Services Key Management Service. https://aws.amazon.com/kms/faqs/

[21] Microsoft Azure. (2024). *About Keys - Azure Key Vault*. Azure Key Vault | Microsoft Learn. https://learn.microsoft.com/en-us/azure/key-vault/keys/about-keys