Enabling Serverless Sky Computing

Robert Cordingly, Wes Lloyd
School of Engineering and Technology
University of Washington
Tacoma, Washington USA
rcording, wlloyd@uw.edu

***Abstract*—The Sky Computing vision represents a unified multi-cloud environment where applications can be deployed to utilize resources from different cloud regions, resource configurations, and cloud providers. Serverless computing platforms have recently emerged, offering automatic elastic scaling, high performance, and reduced costs but often utilize proprietary deployment tools and services locking users into platform-specific services. This research aims to apply Sky Computing to serverless computing platforms offered by major cloud providers such as Amazon Web Services, Google Cloud, Azure, and more. This research will build a serverless sky architecture to enable the aggregation of serverless resources to achieve service-level objectives such as low hosting costs, high performance, fault tolerance, high throughput, and low carbon footprint. The research will focus on evaluating the performance implications of serverless aggregation (Thrust-1), design and evaluation of Sky Computing architectures and aggregation strategies (Thrust-2), and finally autonomous resource aggregation for intelligent self-management of applications deployed to the sky (Thrust-3).**

I. Introduction

The advent of cloud computing has allowed software engineers and developers to create globally distributed applications with essentially unlimited computational resources. Serverless Function-as-a-Service (FaaS) computing platforms offer a streamlined delivery model where cloud resources are automatically managed and scaled to meet user demand, all while ensured with high availability and fault tolerance. Although these platforms simplify application deployment, there are still many challenges and considerations that developers must make to optimize their applications.

To fully utilize serverless FaaS platforms, developers are incentivized to deploy their applications as many independent microservices [1]. Multi-function deployments enable applications composed of many functions that can be elastically scaled independently to meet user demand while more optimally allocating cloud infrastructure. If certain functions are used infrequently, their host infrastructure can be terminated to not incur any hosting costs.

Alongside application composition, serverless computing platforms introduce additional challenges related to vendor lock-in and portability. Many platforms use unique, often proprietary, tools for packaging and deploying applications to the cloud. The lack of standardized deployment processes and the use vendor specific services/libraries make an application deployed to one cloud provider often only compatible with another with extensive code or packaging changes. Alongside that, cloud platforms usually function with a per-region point of view, where cloud resources must be deployed to a single region at a time, potentially resulting in limited interoperability with resources in other regions. While FaaS platforms automatically manage servers to be serverless, they are not yet region-less. The lack of up-front hosting costs and microservice design architecture make it beneficial for developers to deploy functions to every region to reduce latency for users worldwide.

To address these challenges and enable developers to create multi-function, multi-cloud serverless applications easily, this research will investigate and develop techniques to bring Sky Computing to serverless computing platforms. Sky computing involves creating a software compatibility layer that lies above individual cloud providers [2]–[4]. This compatibility layer can enable developers to create large application deployments to many regions worldwide across multiple cloud providers. By aggregating large quantities of deployments using a sky layer, applications can be optimized in new ways that benefit not only the cloud provider, but also the developer, users, and the environment.

II. Related Work

Sky Computing was first discussed in the early 2010s as a means to avoid vendor lock-in on proprietary cloud platforms [5], [6]. Cloud users called for standardization of cloud platforms allowing greater portability and interoperability between resources on different cloud platforms. Tools now exist, such as Docker, Apache libCloud and jClouds, that aid in improving software portability between cloud providers [7], [8]. Although, modern serverless computing platforms usually increase vendor lock-in compared to IaaS platforms as functions use proprietary libraries and deployment tools, reintroducing the need and demand to harness and innovate Sky Computing as a potential solution.

In the last few years, Sky Computing has reemerged. Software compatibility layers that lie above individual cloud providers are being investigated to provide a unified architecture for accessing resources on multiple cloud providers [3]. Yunhao Mao discussed creating SkyBridge, a data management system allowing multi-cloud data storage [4]. Yang et al. created SkyPilot, an intercloud Broker for large language model training and machine learning workloads where workloads are dynamically moved across available cloud providers to reduce cost and increase availability. Sky Computing compatibility layers can be applied to multiple cloud delivery models [9]. A serverless FaaS Sky Computing platform can enable developers to create large application deployments to

many regions worldwide across multiple cloud providers. By aggregating large quantities of deployments using a sky-layer, applications can be optimized in new ways that benefit not only the cloud provider but also the developer, users, and the environment.

III. Research Questions

This proposed research will investigate the aggregation of serverless resources across multiple regions, cloud providers, and deployment configurations to enable Sky Computing concepts for serverless computing platforms. The research will be broken into three thrusts: (Thrust-1) initial FaaS resource aggregation evaluation, (Thrust-2) sky-layer development and trade-off analysis, and finally, (Thrust-3) autonomous application aggregation to enable dynamic application composition, deployment, and management to meet a variety of service-level objectives.

A. Thrust-1: FaaS Resource Aggregation Evaluation

While cloud providers encourage users to decouple their applications into many individual function deployments to achieve optimal elastic scaling, creating large deployments of different configurations can have additional benefits to the user. For example, to offer high availability and low latency for users across the world, developers can deploy their applications to as many regions as possible. Alternatively, multi-region resource aggregation can be used to optimize an application's energy footprint [10], [11], improve fault tolerance, and enable elastic scaling beyond a single region or user account. Each cloud provider offers serverless platforms with varying pricing models and underlying infrastructure. Without extensive testing, developers are left to make ad hoc decisions and may select a platform without knowing if another cloud provider can enable a better price-to-performance outcome [12]–[14].

Developers are responsible for individually deploying each serverless resource in their application. If a developer wanted to take full advantage of an aggregate combination of serverless computing platforms, they would likely need to deploy each serverless resource with dozens, if not hundreds, of configurations to achieve all of the benefits. The primary research question of this thrust is:

(RQ-1): How can serverless resource aggregation optimize for performance objectives such as runtime, latency, throughput, carbon intensity, and cost while ensuring portability and observability?

B. Thrust-2: Sky-layer Development and Trade-off Analysis

After evaluating the potential for serverless resource aggregation, the next direction of our research is to develop sky-layer architecture to increase the scale of our analysis to multi-cloud configurations and learn more about design trade-offs and challenges for aggregating serverless platforms.

Our new sky-layer must aim to provide a platform-neutral architecture that supports the entire serverless application management life-cycle from deployment, application composition, and request routing. The sky-layer would enable users to develop applications and deploy them to different cloud platforms seamlessly; the sky-layer would package the application and communicate with cloud providers through each platform-specific API. Since the sky-layer is essentially abstracting away the cloud provider, careful consideration must be made for each cloud provider's unique feature set. A platform-neutral format must be maintained while providing access to all of a platform's available features. Varying features and potential platform incompatibility would create a granularity where some applications could be aggregated with resources among various configurations (enabled course-grained aggregation). In contrast, others would be more fine-grained and only aggregate with resources in the same region or cloud provider. The trade-off between fine-grained aggregation and course grained could lead to performance or cost variability.

The sky-layer would enable the aggregation of serverless resources across multiple configurations and cloud providers and offer a streamlined approach to application management. Using the sky-layer this research will evaluate different resource aggregation strategies, such as multi-configuration aggregation, multi-region, and multi-cloud aggregation, and assess trade-offs in performance, latency, costs, scalability, and fault tolerance with varying granularity of resource aggregation. Thrust-2 will investigate the following research questions:

(RQ-2): How can platform-neutral abstractions be innovated to improve compatibility between platforms while providing feature parity on serverless cloud platforms?

(RQ-3): What are the trade-offs of different resource aggregation and deployment strategies (e.g. multi-region, multi-cloud, multi-configuration deployment) utilized in the sky-layer?

C. Thrust-3: Autonomous Application Aggregation

The third thrust of this research will utilize the newly created sky-layer to autonomously compose and aggregate serverless resources to meet the goals evaluated in Thrust-1. Our previous publications used machine learning techniques such as linear regression, multiple regression, and random forest to predict the performance of one serverless configuration based on another [12]. We also developed the CPU Time Accounting Memory Selection model to predict optimal memory configurations for serverless functions with minimal profiling data [15].

Thrust-3 will apply lessons from prior work with the sky-layer created in Thrust-2 to create an autonomous sky computing application composition system. This will enable applications to be developed in the sky-layer and then intelligently deployed across multiple cloud providers to meet service level objectives such as low latency, low cost, low carbon footprint, and more. The sky-layer will leverage aggregations of serverless resources and intelligently self-manage applications to fit user demand and abstract additional configuration details of existing FaaS platforms to deliver a serverless Sky Computing platform. For example, the sky-layer will be able to relocate functions to different regions or cloud providers based on user

demand or the carbon intensity of a region. If an application receives requests requiring more hardware resources than what is available in an aggregation, the sky-layer can create new serverless resources to meet the demand or offer a more desirable price-to-performance ratio. Thrust-3 will focus on evaluating the research question:

(RQ-4): How can serverless aggregation strategies be dynamically learned and applied for specified goals (reduced cost, reduced latency, reduced carbon footprint) for different aggregations of serverless resources (e.g., multi-region, multi-cloud, multi-configuration)?

IV. Experimental Methodology and Expected Contributions

To begin the research, develop a proof of concept, and acquire preliminary results, we made large application deployments to many regions with gradually increasing complexity. We deployed 12 different serverless functions to 19 regions on AWS Lambda for our initial tests. We then executed experiments over six months to observe the performance variability of these regions in terms of function runtime and network latency. Using an intelligent proxy function to route requests, we observed the potential for latency and carbon intensity improvements by aggregating serverless resources. Thrust-1 will continue to grow the complexity of this experiment to expand and evaluate multi-configuration and multi-cloud function deployments. The Function-as-a-Service Experiment Toolkit (FaaSET) has the functionality to make FaaS platform neutral functions with improved portability and compatibility with multiple cloud providers [16], which can be used for serverless platform evaluation before developing the sky-layer in Thrust-2. This research will continue long-term multi-month experiments where data is collected for performance variability of multiple cloud providers and varying deployment configurations (RQ-1).

Thrust-1 and RQ-1 validate the applicability of Sky Computing on serverless platforms before moving to Thrust-2. Thrust-2 will begin developing the sky-layer and shift focus from performance analysis in Thrust-1 to architectural design and trade-off analysis in Thrust-2. RQ-2 focuses on the design of the sky-layer architecture. As a proof of concept, we created a serverless proxy function to route requests between multiple regions and optimize for carbon intensity as shown in Figure 1.

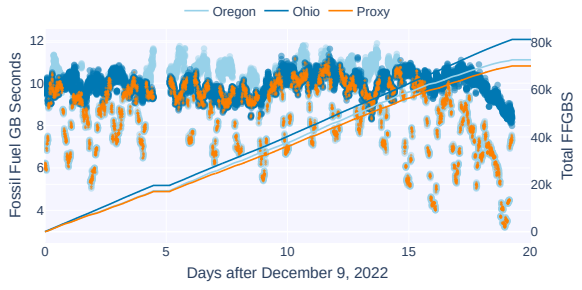

Fig. 1. Carbon intensity reduction using serverless aggregation and a smart proxy to distribute requests between two regions in North America.

The sky-layer will consist of many components, such as application packaging, deployment, resource management, routing/scheduling, resource monitoring, and more. The intelligent proxy is a proof of concept for the routing/scheduling component of the sky-layer. Thrust-2 will evaluate multiple implementations of each component to evaluate design and performance trade-offs (RQ-2). After the sky-layer is implemented, it will be utilized to build and aggregate a wide variety of serverless resources with varying degrees of granularity to investigate trade-offs and performance implications of aggregation itself (RQ-3). Experiments will evaluate different aggregations; for example, an aggregation composed of one set of cloud providers may offer differing latency compared to another cloud provider based on the number of regions and locations for users distributed throughout the world. Different cloud providers offer varying sets of back-end infrastructure and include slightly different pricing models, so the cost-to-performance ratio of a workload may vary by the cloud provider. For workloads that require extremely high throughput, aggregating resources across cloud providers can increase the amount of accessible hardware, improving the performance of the workload.

Thrust-3 will expand on the sky-layer and focus on enabling intelligent self-management of serverless aggregations. The focus will be to develop models and heuristics for autonomous resource management and aggregation from the profiling experiments in Thrust-1 and the trade-off analysis in Thrust-2. Our previous publications focused on performance modeling and autonomous function configuration [15], [17]. Thrust-3 will apply the previous models and train new ones to autonomously configure and update serverless aggregations to meet new or changing service-level objectives. At the sky-layer, applications could be assigned different SLOs, such as minimal latency, and automatically deploy to regions that are located nearby the users of the application. Data from the experiments in Thrust-2 can be used to train models and evaluate their effectiveness at meeting set SLOs.

A. Preliminary Results

- In our initial evaluation of FaaS resource aggregation, we evaluated the performance and latency of every region on AWS Lambda for six months. We found that latency had a coefficient of variation between 2-29% during the day, varying on average +/-10 ms. Function runtime varied much less, with 3 to 6% CV across various workloads (RQ-1).
- In our initial multi-region aggregation experiment, we compared workloads deployed in a single region to a multi-region aggregation utilizing 19 regions. We were able to reduce latency by, on average, 65% while reducing the carbon intensity by up to 99.8% compared to an application deployed to a single region.
- We have developed initial tools that can be built upon to create the sky-layer of a serverless resource aggregation system. Components include the Serverless Application

Analytics Framework (SAAF) and the Function-as-a-Service Experiment Toolkit (FaaSET) (RQ-2 and RQ-3).
- We developed the CPU Time Accounting Memory Selection (CPU-TAMS) model. Using a single profiling run, CPU-TAMS was shown to predict function memory configurations offering the best price-to-performance ratio with only 5% cost and 8% runtime error on AWS Lambda. CPU-TAMS required significantly less profiling data, resulting in a 3.8 to 15x lower cost to apply compared to exhaustive search techniques (RQ-4).
- By deploying a function with multiple different memory configurations (multi-configuration aggregation), we were able to leverage the CPU-TAMS model [15] to reduce function hosting costs by 58% (RQ-4).

V. Conclusions

Sky Computing concepts applied to serverless computing platforms have the potential to create a serverless multi-cloud computation ecosystem. By aggregating serverless resources across multiple cloud regions, resource configurations, and cloud providers, applications can be optimized for high performance, low latency, high throughput, and low cost while reducing the total energy footprint.

The proposed research will be organized into three research thrusts. Thrust-1 evaluates the benefits of aggregating serverless resources to meet various service-level objectives using long-term experiments over multiple months to observe performance variability (RQ-1). Thrust-2 begins the development of the Sky-layer and the components to facilitate streamlined serverless resource aggregation. This research will evaluate multiple architecture implementations (RQ-2) and investigate different levels of abstraction for platform-neutral serverless applications (RQ-3). After the sky-layer has been implemented, Thrust-3 expands upon it to achieve intelligent self-management of serverless aggregations. Thrust-3 will develop and train models to autonomously create and reconfigure aggregations to account for changing SLOs (RQ-4).

The preliminary experiments and previous research show the potential benefits of serverless aggregation, including reduced latency, improved performance, reduced costs, and significantly reduced carbon footprint [18].

VI. Biography

Robert Cordingly is a Ph.D. student studying Distributed Systems and Cloud Computing in the Schools of Engineering and Technology at the University of Washington. Robert's research is supported by the NSF Office of Advanced Cyberinfrastructure and AWS Cloud Credits for Research. He has collaborated with various organizations, including Biodepot LLC, where he deployed serverless bioinformatics workloads, and the USDA Agricultural Research Service, where he's working on streamlining particle swarm optimization models on the cloud. Robert aims to graduate in the spring of 2025.

References

[1] L. Wang, M. Li, Y. Zhang, T. Ristenpart, and M. Swift, "Peeking Behind the Curtains of Serverless Platforms," *2018 USENIX Annual Technical Conf. (USENIX ATC 18)*, 2018.

[2] S. Chasins, A. Cheung, N. Crooks, A. Ghodsi, K. Goldberg, J. E. Gonzalez, J. M. Hellerstein, M. I. Jordan, A. D. Joseph, M. W. Mahoney *et al.*, "The sky above the clouds," *arXiv:2205.07147*, 2022.

[3] I. Stoica and S. Shenker, "From cloud computing to sky computing," in *Proceedings of the Workshop on Hot Topics in Operating Systems*, 2021, pp. 26–32.

[4] Y. Mao, "Skybridge: A cross-cloud storage system for sky computing," in *23rd International Middleware Conference Doctoral Symposium*. ACM, 2022, pp. 15–17.

[5] D. Petcu, C. Crăciun, M. Neagul, S. Panica, B. Di Martino, S. Venticinque, M. Rak, and R. Aversa, "Architecturing a sky computing platform," in *Towards a Service-Based Internet. ServiceWave 2010 Workshops: International Workshops, OCS, EMSOA, SMART, and EDBPM 2010, Ghent, Belgium, December 13-15, 2010, Revised Selected Papers 3*. Springer, 2011, pp. 1–13.

[6] K. Keahey, M. Tsugawa, A. Matsunaga, and J. Fortes, "Sky computing," *IEEE Internet Computing*, vol. 13, no. 5, pp. 43–51, 2009.

[7] I. Docker, "Docker," *línea].[Junio de 2017]. Disponible en: https://www. docker. com/what-docker*, 2020.

[8] "Apache libcloud," 2023. [Online]. Available: https://libcloud.apache.org

[9] Z. Yang, Z. Wu, M. Luo, W.-L. Chiang, R. Bhardwaj, W. Kwon, S. Zhuang, F. S. Luan, G. Mittal, S. Shenker *et al.*, "Skypilot: An intercloud broker for sky computing," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 437–455.

[10] N. Bashir, T. Guo, M. Hajiesmaili, D. Irwin, P. Shenoy, R. Sitaraman, A. Souza, and A. Wierman, "Enabling sustainable clouds: The case for virtualizing the energy system," in *ACM Symposium on Cloud Computing (SoCC)*, 2021.

[11] R. Farahani, D. Kimovski, S. Ristov, A. Iosup, and R. Prodan, "Towards sustainable serverless processing of massive graphs on the computing continuum," in *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*. ACM, 2023, pp. 221–226.

[12] R. Cordingly, W. Shu, and W. J. Lloyd, "Predicting Performance and Cost of Serverless Computing Functions with SAAF," in *6th IEEE Int. Conf. on Cloud and Big Data Computing (CBDCOM 2020)*, 2020.

[13] R. Cordingly, N. Heydari, H. Yu, V. Hoang, Z. Sadeghi, and W. Lloyd, "Enhancing observability of serverless computing with the serverless application analytics framework," in *Companion of the 2021 ACM/SPEC Int. Conf. on Performance Engineering, Tutorial*, 2021.

[14] R. Cordingly, H. Yu, V. Hoang, Z. Sadeghi, D. Foster, D. Perez, R. Hatchett, and W. Lloyd, "The serverless application analytics framework: Enabling design trade-off evaluation for serverless software," in *Proc of the 2020 Sixth Int. Workshop on Serverless Computing*, 2020, pp. 67–72.

[15] R. Cordingly, S. Xu, and W. Lloyd, "Function memory optimization for heterogeneous serverless platforms with cpu time accounting," in *2022 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 2022, pp. 104–115.

[16] R. Cordingly and W. Lloyd, "Faaset: A jupyter notebook to streamline every facet of serverless development," in *Companion of the 2022 ACM/SPEC International Conference on Performance Engineering*, 2022, pp. 49–52.

[17] R. Cordingly, "Serverless performance modeling with cpu time accounting and the serverless application analytics framework," 2021.

[18] R. Cordingly, J. Kaur, D. D., and W. Lloyd, "Towards federated serverless computing: An investigation on global workload distribution to mitigate carbon intensity, network latency, and cost," in *2023 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 2023.