Modeling a Multi-Objective Optimization Problem for Sustainable Serverless Computing

1st Jaehwan Lee
Department of Computer Science and Engineering
Kongju National University
Cheonan, Korea (South)
jhnlee@kongju.ac.kr

2nd Yeunwoong Kyung
Department of Electronic Engineering
Seoul National University of Science and Technology
Seoul, Korea (South)
ywkyung@seoultech.ac.kr

3rd Won Young Jeon
Department of Computer Science and Engineering
Kongju National University
Cheonan, Korea (South)
jun1young@kongju.ac.kr

4th Sangoh Park
School of Computer Science and Engineering
Chung-Ang University
Seoul, Korea (South)
sopark@cau.ac.kr

Abstract—Sustainable cloud computing aims to reduce the environmental impact of growing computing demand. Serverless computing can improve efficiency through on-demand execution of service requests. However, existing studies prioritize either performance or energy constraints, overlooking the trade-offs between carbon footprints and cold-start latency. This paper explores a multi-objective optimization problem that can balance these conflicting goals. The deisgn will provide a theoretical foundation for learning policies that optimize performance with environmental sustainability.

Index Terms—serverless computing, cloud computing, sustainable computing

I. Introduction

Recently, we experience an exponential increase in computing demand worldwide. Large-scale data analytics and internet of everything paradigm have increased data center power consumption and greenhouse gas emissions [1]. Sustainable cloud computing aims to mitigate this environmental impact by optimizing resource usage. Serverless computing has emerged as a key to address these challenges [2]. Unlike traditional server-based models that require provisioning with idle resource waste, serverless architectures enable on-demand resource allocation. This can offer potential to reduce energy consumption and improve computational efficiency by allocating and reclaiming resources upon request, completion, respectively [3].

However, effective resource management remains difficult due to stochastic and bursty workloads. The cold-start problem is caused by reclaiming resources during idle periods. It induces not only latency but also energy overheads from frequent instance warm-ups [4]. Therfore, proactive provisioning and energy-aware scheduling have emerged for sustainable serverless computing [5].

In this paper, we investigate the potential of sustainable serverless computing by jointly addressing dynamic carbon footprints and cold-start latency. Through an analysis of existing studies [4] [5] [6] [7] [8], we identified the trade-offs between performance and environmental footprint, and that the existing studies lack of consideration in both carbon footprint and cold-start latency. Based on this analysis, we aim to derive a strategy that ensures service latencies while minimizing carbon emissions.

II. Problem Formulation

We consider a serverless platform operating over a discrete time horizon $\mathcal{T}$. The platform manages a set of distinct serverless functions $\mathcal{F}$. The arrival rate of invocations for function $f \in \mathcal{F}$ at time slot t is denoted by $\lambda_f(t)$. The environmental impact is governed by the carbon intensity of the electricity grid, $CI(t)$, measured in gCO_2eq/kWh. $CI(t)$ varies temporally based on the type of energy (e.g., renewable vs. fossil fuels) Then the platform dynamically manages the lifecycle of function instances: cold, warm, or active. If an incoming request arrives and a warm instance is available, this incurs a minimal latency, L_W. If no warm instances are available, a cold-start latency is required, L_C. Upon completion of execution, an instance transitioning from active to warm remains for a duration by the keep-alive policy, $T_{keep,f}(t)$, before being terminated if no new requests arrive. In addition, the controller can proactively initialize instances, $N_{PW,f}(t)$, in anticipation of future demand or favorable carbon conditions.

Let $N_f^A(t)$, $N_f^W(t)$ denote the number of active and warm instances at time t, respectively, for function f at the beginning of time slot t. The number of cold starts for function f during time slot t, $N_{cold,f}(t)$, occurs when the demand exceeds $N_f^W(t)$. Then the avrerage response time for f at t, incorporates the execution time $T_{exec,f}$:

$$ART_f(t) = \frac{N_{cold,f}(t)}{\lambda_f(t)} \cdot (L_C + T_{exec,f}) + \frac{\lambda_f(t) - N_{cold,f}(t)}{\lambda_f(t)} \cdot (L_W + T_{exec,f}) \quad (1)$$

The overall system performance penalty at t is defined as:

$$\mu_{perf}(t) = \sum_{f \in \mathcal{F}} w_f \cdot ART_f(t) \quad (2)$$

where w_f represents the priority weight of function f.

The execution energy is determined by the total workload processed:

$$E_{exec}(t) = \sum_{f \in \mathcal{F}} \lambda_f(t) \cdot T_{exec,f} \cdot P_{active} \quad (3)$$

The initialization energy is incurred by both reactive cold starts and proactive provisioning:

$$E_{init}(t) = \sum_{f \in \mathcal{F}} (N_{cold,f}(t) + N_{prov,f}(t)) \cdot E_{init\_cost} \quad (4)$$

The idle energy is determined by the number of warm instances:

$$E_{idle}(t) = \sum_{f \in \mathcal{F}} \overline{N_f^W}(t) \cdot P_{idle} \quad (5)$$

The total operational carbon emission at t is defined as:

$$C_{total}(t) = (E_{exec}(t) + E_{init}(t) + E_{idle}(t)) \cdot CI(t) \quad (6)$$

Therefore, the objective is formulated as follows. The objective is to develop an optimal control policy π that dynamically adjusts the function lifecycle management to minimize the long-term cumulative cost. The cost function $J(t)$ balances the trade-off between performance degradation and carbon emissions:

$$J(t) = \alpha \cdot \mu_{perf}(t) + \beta \cdot C_{total}(t) \quad (7)$$

where α and β are hyperparameters that weigh the relative importance of performance and sustainability objectives.

Finally, the objective can be formulated as minimizing the expected cost over $\mathcal{T}$:

$$\min_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t J(t)\right] \quad (8)$$

where $\gamma \in [0, 1)$ is the discount factor.

III. Conclusion

We introduced a multi-objective optimization for sustainable serverless computing, modeling trade-offs between performance degradation and carbon emissions. Then we formulated a problem that captures the dynamics of serverless function lifecycle and energy consumptions. By defining a cost function, we could establish a theoretical foundation for optization of sustainable serverless computing based on deep reinforcement learning.

Our future work will focus on solving this formulated optimization problem. We plan to design and implement a Deep Reinforcement Learning (DRL) approach to derive an adaptive control policy that can dynamically navigate the trade-offs in real-time. Furthermore, we will validate the effectiveness of the learned policy through real-world trace simulations, comparing its performance against existing performance-centric and static carbon-aware strategies.

References

[1] U. Gupta et al., “Chasing Carbon: The Elusive Environmental Footprint of Computing,” in Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA), 2021, pp. 854-867.

[2] I. Baldini et al., “Serverless Computing: Current Trends and Open Problems,” in Research Advances in Cloud Computing, S. Chaudhary et al., Eds. Singapore: Springer, 2017, pp. 1–20.

[3] Y. S. Patel and P. Townend, “A Stable Matching Approach to Energy Efficient and Sustainable Serverless Scheduling for the Green Cloud Continuum,” in Proc. IEEE Int. Conf. Service-Oriented Syst. Eng. (SOSE), Apr. 2024, pp. 25-35.

[4] Y. Jiang, B. Li, R. Basu Roy, and D. Tiwari, “ECOLIFE: Carbon-Aware Serverless Function Scheduling for Sustainable Computing,” in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC), Nov. 2024.

[5] Y. Hu, J. Yang, X. Ruan, Y. Chen, C. Li, Z. Zhang, and W. Zhang, “Green optimization for micro data centers: Task scheduling for a combined energy consumption strategy,” Applied Energy, vol. 393, p. 126031, 2025.

[6] J. Serenari, S. Sreekumar, K. Zhao, S. Sarkar, and S. Lee, “GreenWhisk: Emission-Aware Computing for Serverless Platform,” in Proc. IEEE Int. Conf. Cloud Eng. (IC2E), Mar. 2024, pp. 44-54.

[7] L. Gu, W. Zhang, Z. Wang, D. Zeng, and H. Jin, “Service Management and Energy Scheduling Toward Low-Carbon Edge Computing,” IEEE Trans. Sustain. Comput., vol. 8, no. 1, pp. 109-119, Jan.-Mar. 2023.

[8] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, “Let’s Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud,” in Proc. 22nd Int. Middleware Conf., Dec. 2021, pp. 260-272.