M2 PDS Research Project | Starting 2026-10-12

Powerful Serverless: Power Measurement of Serverless Functions

Mathieu Bacou
SAMOVAR, Télécom SudParis, Inria, Institut Polytechnique de Paris, France
mathieu.bacou@telecom-sudparis.eu

François Trahay
SAMOVAR, Télécom SudParis, Inria, Institut Polytechnique de Paris, France
francois.trahay@telecom-sudparis.eu

Keywords: power usage, Serverless, FaaS, cloud, systems, measurements

I. Context

Cloud computing is now a major foundation of digital infrastructures. Among its service models, *Function-as-a-Service*[1] (FaaS) the automatic execution of software functions in response to events, without requiring the client to manage the allocation of the necessary hardware and software infrastructure resources. This model is characterized by the huge number of instances of those functions, the fast rate they are created, used and destroyed, and their quick individual execution time. In addition to this scale problem, observing FaaS platforms and measuring metrics to characterize their execution is a challenge due to the complexity of their software stacks: function instances are materialized as isolated software environments using combinations of virtualization techniques.

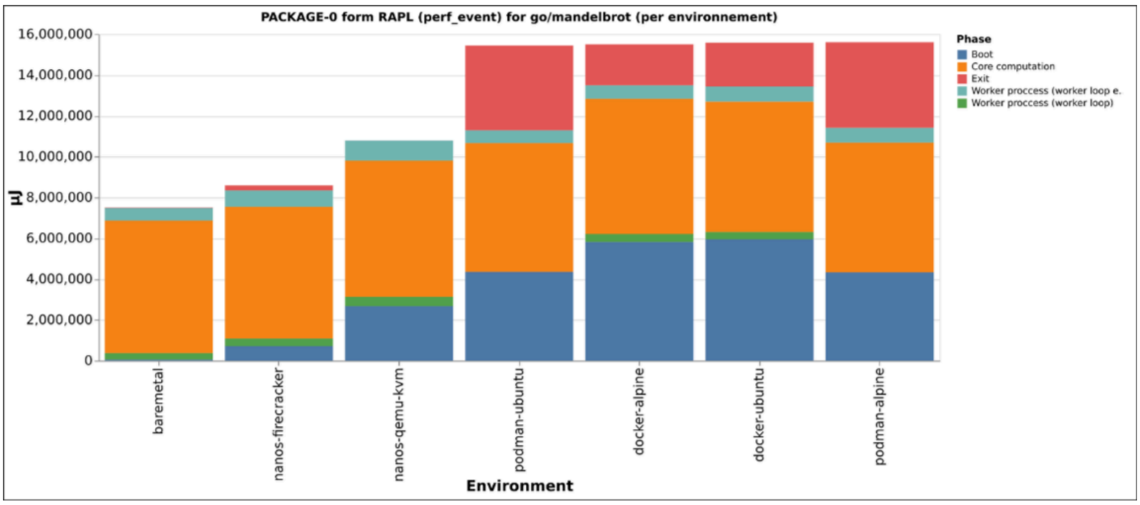

Figure 1: Power usage of a function "Mandelbröt" under various environments, breakdown by FaaS execution phase.

It makes it hard to understand a critical part of their behavior on the systems: their power usage. Indeed, existing power meter solutions are not fit for the properties of FaaS[2], [3], [4]. Beyond the most basic meters that only measure the global consumption of power of the physical machines, *software-defined* power meters try to give meaningful information about applications. Most often, they rely on some previous training that requires a lot of data and time. This limit makes them largely impractical, especially in FaaS environments, because the requirement is to understand the power usage behavior of user-provided workloads that is by definition unknown at first.

The Powerful Serverless project takes place within the nation-wide collaborative project EVERGREEN, specifically its Work Package of developing *tools to precisely measure and optimize the energy consumption* of a FaaS platform. Here, we aim to (i) measure energy consumption at the granularity of a single function instance to inform users of their usage, and (ii) manage function execution to minimize energy consumption at the platform scale. The targeted challenges are:

1. Measuring raw metrics at the platform's system level;
2. Interpreting this data at scale and within the context of the FaaS platform;
3. Implementing actionable levers for energy efficiency based on this data.

The Powerful Serverless project starts by focusing on the first challenge.

II. Goals

EVERGREEN and Powerful Serverless build on PowerAPI[5], a state-of-the-art software wattmeter developed by EVERGREEN contributors. The extensibility of PowerAPI will provide a solid foundation for modeling, indicators, and analysis of FaaS platforms. PowerAPI already enables the collection of energy consumption measurements from hardware sensors. However, its design was *not intended for the scale and execution time* characteristics of serverless function instances.

The goals of Powerful Serverless are:

- Identify the design constraints of software systems (virtualization, Linux kernel perf subsystem, etc.) and hardware (RAPL interface) related to energy measurement that pose challenges in a serverless context;
- Propose new designs for software layers (including PowerAPI) and identify their limitations due to hardware constraints;
- Experimentally validate these solutions.

Further objectives include researching the data interpretation challenge, to model the power usage of FaaS applications, functions and hosts.

III. Work environment

The student will integrate into the research team 🗗 Benagil of Inria, who studies how to design more efficient and secure distributed systems by focusing on their core components.

Technology-wise, the student(s) will gain experience in bleeding-edge cloud infrastructures, with FaaS platforms (mainly Apache OpenWhisk), cluster management (containers, Kubernetes), and hardware virtualization (virtual machines) applied to state-of-the-art issues of the cloud. Development can be expected in Python (for PowerAPI) as well as systems language (C for PowerAPI and work in the Linux kernel, possibly Go for containerization layers, otherwise left to the students' preference).

Bibliography

[1] M. Shahrad *et al.*, "Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider," in *Proceedings of the 2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, A. Gavrilovska and E. Zadok, Eds., USENIX Association, 2020, pp. 205–218.

[2] J. Treibig, G. Hager, and G. Wellein, "LIKWID: A Lightweight Performance-Oriented Tool Suite for x86 Multicore Environments," in *39th International Conference on Parallel Processing, ICPP Workshops 2010, San Diego, California, USA, 13-16 September 2010*, W.-C. Lee and X. Yuan, Eds., IEEE Computer Society, 2010, pp. 207–216. doi: 10.1109/ICPPW.2010.38.

[3] M. Amaral *et al.*, "Process-Based Efficient Power Level Exporter," in *17th IEEE International Conference on Cloud Computing, CLOUD 2024, Shenzhen, China, July 7-13, 2024*, R. N. Chang, C. K. Chang, J. Yang, N. L. Atukorala, Z. Jin, M. Sheng, J. Fan, K. Fletcher, Q. He, T. Kosar, S. Sarkar, S. Venkateswaran, S. Wang, X. Liu, S. Seelam, C. Narayanaswami, and Z. Zong, Eds., IEEE, 2024, pp. 456–467. doi: 10.1109/CLOUD62652.2024.00058.

[4] R. Brondolin, T. Sardelli, and M. D. Santambrogio, "DEEP-Mon: Dynamic and Energy Efficient Power Monitoring for Container-Based Infrastructures," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPS Workshops 2018, Vancouver, BC, Canada, May 21-25, 2018*, IEEE Computer Society, 2018, pp. 676–684. doi: 10.1109/IPDPSW.2018.00110.

[5] G. Fieni, D. R. Acero, P. Rust, and R. Rouvoy, "PowerAPI: A Python framework for building software-defined power meters," *J. Open Source Softw.*, vol. 9, no. 98, p. 6670, 2024, doi: 10.21105/JOSS.06670.