Project Title: Serverless Workload Classification

Background and Motivation

In serverless Function-as-a-Service (FaaS), cloud providers handle resource management for each function (e.g., scaling the number of function containers or their resource limits and scheduling containers to servers), but existing commercial cloud providers provide no performance guarantees such as service-level objectives (SLOs). Providing performance SLOs has been studied in various aspects and is critical to running latency-critical services on serverless platforms. The problem of managing resources to achieve performance SLOs while maintaining high resource utilization is, at its core, an intractable NP-hard problem. While the majority of the associated problems are approached using meticulously designed heuristics with extensive application-/system-specific domain-expert-driven tuning, a substantial line of work has recently been focused on learning-based approaches such as reinforcement learning (RL).

However, a learned RL policy is application workload-specific. Substantial **retraining** is needed to adapt to new workloads in heterogeneous and dynamically evolving (possibly multi-cloud) datacenters. For instance, application performance and utilization differ significantly among heterogeneous workloads. Consequently, a trained RL policy suffers performance degradation and requires substantial retraining, even with transfer learning, to adapt to new workloads. It is thus a critical problem in making RL practical in production.

Goal. In this project, we aim to start with serverless workload classification and design a classifier to group application workloads into several major categories. Instead of training one model/policy per application workload, we can train ("**pre-train**") several foundation models per classified category. After that, only lightweight **fine-tuning** is needed for fast adaptation to learn customized models/policies.

Project Milestones

Week 1-2 09/06-09/19

Understand basic concepts in:

- Serverless computing and resource management
  - https://www.usenix.org/system/files/atc20-shahrad.pdf
  - https://arxiv.org/pdf/2009.08173.pdf
  - https://arxiv.org/pdf/2105.11592.pdf
- Learning-based approaches for resource management
  - https://people.csail.mit.edu/alizadeh/papers/deeprm-hotnets16.pdf
  - https://haoran-qiu.com/pdf/firm.pdf
  - https://dl.acm.org/doi/pdf/10.1145/3510415
- Classification
  - Supervised learning-based approach (not tenable in our setting!): https://webapps.cs.umu.se/uminf/reports/2013/013/part1.pdf
  - Characterization of Hadoop jobs with unsupervised learning: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5708526

    - Paragon - Collaborative filtering: https://dl.acm.org/doi/pdf/10.1145/2499368.2451125
    - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8721402

Week 3-4 09/20-10/03

Get familiar with the serverless dataset:

- Understand the fields/columns of a serverless dataset
- Estimate the CPU and memory utilization based on performance counters
    - CPU Util = (sysDiff + userDiff) / 1000 / duration * 100%
    - Memory Util = heapUsed / heapTotal * 100%
- Investigate the relationship between resource allocation vs. performance
- Visualization (Open Task)

Week 5-6 10/04-10/17

Heuristics-based classification:

- Dataset cleaning and preprocessing
- Explore and pick any potentially critical heuristics/metrics for classification, e.g., CPU-intensive vs. memory-intensive, high heap usage vs. low heap usage, sensitivity to any allocation changes, etc.

Week 7-8 10/18-10/31

ML-based classification:

- Explore and evaluate different unsupervised learning algorithms for classification
- Explore contrastive learning if possible
- Visualization of the classification results

Week 9-10 11/01-11/14

Validation

- Run new applications (from selected serverless benchmarks) on a serverless platform
- Validate the developed and trained ML models
- Optional: Validate the classified workload clusters in a use case of an RL-based autoscaler (by checking if less adaptation is required for workloads within the same cluster)

Week 11-12 11/15-11/30

Wrap-up

- Report
- Presentation
- Future work