Poster Abstract: Automatic Deployment Right-Sizing Through Hyperparameter Optimization

Aniruddha Rakshit
Binghamton University
Binghamton, New York, USA

Jayson Boubin
Binghamton University
Binghamton, New York, USA

ABSTRACT

Internet of Things (IoT) and Edge deployments are diverse, complex, and highly constrained. These properties make correctness difficult or impossible to verify a priori. We present early work on an automatic deployment right-sizing tool for edge and IoT deployments. Our tool uses the PROWESS testbed to accurately emulate candidate deployment form-factors, and optimizes deployment parameters to minimize costs. We show that our early work finds optimal deployment configurations 6.3X faster than Bayesian optimization, a state of the art hyperparameter optimization technique.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

Internet of Things, Edge Computing, Rightsizing, UAV

ACM Reference Format:
Aniruddha Rakshit and Jayson Boubin. 2023. Poster Abstract: Automatic Deployment Right-Sizing Through Hyperparameter Optimization. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 9–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3576842.3589157

1 INTRODUCTION

The Internet of Things (IoT) and Edge computing constitute new and expressive levels of the deployment hierarchy for modern workloads [10]. New and interesting sensors are driving innovation, new hardware platforms and accelerators are facilitating new applications, and privacy concerns are growing. Edge computing and IoT answer these challenges by providing sensors with increased intelligence, enhanced security, and rich configurability.

Edge and IoT deployments are, however, difficult to verify in development [5]. It can be unclear whether software, hardware, models, and networks will behave well in concert without deploying in real-world conditions. Recently, edge and cloud testbeds [2, 3, 8] have arisen to provide pre-deployment testing. These platforms allow users to test edge and IoT related software, but are be too remote or virtualized to mimic bare-metal performance. Container-based platforms [3, 7] and PROWESS [3] use local containers that approximate bare-metal performance, but still require users to explore configurations manually. Users must select hardware, software, models, and network parameters for tens or hundreds of devices before deployment. Exploring this parameter space can be prohibitively time consuming for users, resulting in poor configurations that increase costs or fail unexpectedly.

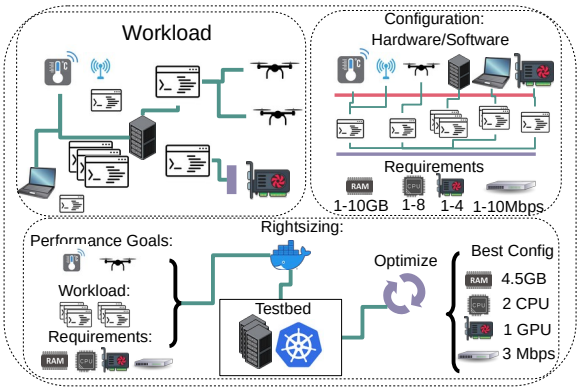

Fig. 1: Workflow for an optimized IoT workload.

We present early work on an automatic deployment right-sizing tool for edge and IoT deployments. Our tool uses a new optimization algorithm based on Pareto Simulated Annealing (PSA) [6] to quickly find both feasible and optimal deployment configurations. Our tool models deployments as a hyperparameter tuning problem, quickly exploring parameter options to find an optimal parameter set. Our tool runs on PROWESS, a container-based edge computing testbed, to run experiments at bare-metal performance. Early results show that our tool can find optimal configurations for deployment 6.3x faster than hyperopt by considering the shape of IoT workloads.

2 DESIGN

Figure 1 shows a conventional IoT workload. Sensors collect data from the world around them, process it in-situ or at the edge, and relay it to a centralized edge or cloud node. Each aspect of this workload requires configuration. Sensors and compute nodes must be correctly provisioned, software must operate within the capabilities of hardware, network properties must be well-understood, and system goals must be well-calibrated. For complex deployments, each of these artifacts and their needs must be tested assure correctness and select appropriate, cost-effective hardware. Users can test workloads in custom testbeds, but these tests only assure correctness. Determining bottlenecks and right-sizing components requires manual parameter tuning. We view this as a hyperparameter tuning problem. Hyperparameters can include available RAM, CPU type and share, network characteristics, AI models, system

IoTDI '23, May 9–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0037-8/23/05…$15.00
https://doi.org/10.1145/3576842.3589157

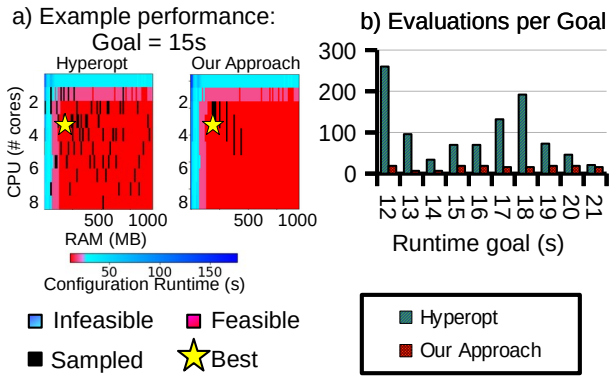

Fig. 2: Early results show that a) our approach quickly finds optimal configurations compered to prior approaches, and b) this finding holds as goals change.

goals, and more. Balancing these characteristics can be difficult. Conventional hyperparameter tuning approaches use sampling to learn the shape of underlying objective functions [1]. Our approach, however, uses intuition about the underlying shape of IoT deployments to converge faster.

Architectural parameters for IoT workloads often improve performance when increased (e.g more CPU, faster runtime), or hurt performance when decreased (E.g slower networks, more missed deadlines). Using this intuition, we devised an optimization algorithm that balances these parameters. Our algorithm, to appear in full in an upcoming publication, is based on Pareto Simulated Annealing (PSA) [6]. PSA uses simulated annealing to find and navigate the Pareto frontier. Given that the optimal configuration is situated on the Pareto frontier, this algorithm should return an optimal configuration in less evaluations than approaches that sample an objective function with no intuition about its underlying shape.

3 EARLY RESULTS

Candidate Workload: For our evaluation, we selected a real IoT workload for Autonomous UAV (AUAV) feature extraction. AUAVs fly through their environments, sense data, and respond in real-time [9]. This response requires online and in-situ processing of sensed data in order to make decisions. Our test workload is a feature extractor for autonomous UAV navigation in soybean fields [4]. This workload takes sensed data from UAVs and runs multiple parallel image processing algorithms to extract features to be fed to reinforcement learning algorithms for pathfinding. This is a critical workload for an autonomous UAV, and assuring its correct execution and proper provisioning is integral to mission success.

Results: We optimized our workload along two axes: RAM and CPU. We wanted to find the feasible configuration that used the least CPU and RAM. Feasibility was determined by the speed at which our workloads accomplished their goal. We ran our sample workload on a PROWESS node with a 12-core Intel Xeon CPU, Nvidia A5000 GPU, and 256 GB of RAM. We ran a containerized version of our workload and used PROWESS to constrain it to various CPU and RAM limits. Our objective function weighted RAM and CPU utilization equally and sought the minimum RAM and CPU combination that finished execution before a set time.

Figure 2 (a) shows our workload when our performance goal was set to 15s. Areas in red denote feasible configurations that finished execution in less than 15s. The magenta boarder of this range represents the Pareto frontier, where workloads begin performing slower than our goal. Blue configurations are infeasible, meaning they take longer than our 15s goal time to execute. In Figure 2 (a), we see the performance of our modified PSA approach as compared to Hyperopt, a state of the art hyperparameter tuning approach. Hyperopt uses Bayesian optimization to build and update a Gaussian prior reflecting the shape of the underlying objective function. This process is reflected by the black regions in the first plot of Figure 2 (a). Here, hyperopt samples our workload and is eventually able to identify the optimal configuration in 70 evaluations.

Our approach exploits the underlying shape of IoT deployments. We start our evaluations at the center of our configuration space. Moving out along both axes in increments based on a set temperature, we evaluate points and select new candidates as performance improves. Given that increased RAM and CPU should garner performance improvement, and feasibility should follow a Pareto frontier, this approach quickly finds and navigates this frontier until an optimal configuration is found. Figure 2 (b) shows that our approach outperforms hyperopt even as goals change. For this workload, our approach finds the optimal configuration in 15.7 evaluations as opposed to hyperopt which takes on average 99.4 evaluations, constituting a 6.3X improvement.

In the future, we plan to evaluate this algorithm on a diverse set of IoT workloads in simulation and deployment. We believe that this tool will help identify scale bottlenecks for IoT deployment configurations, decease deployment costs, and eliminate deployment failures due to misconfiguration.

Acknowledgments: This work was funded by Binghamton University.

REFERENCES

[1] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.
[2] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar. Geni: A federated testbed for innovative network experiments. *Computer Networks*, 61:5–23, 2014.
[3] J. Boubin, A. Banerjee, J. Yun, H. Qi, Y. Fang, S. Chang, K. Srinivasan, R. Ramnath, and A. Arora. Prowess: An open testbed for programmable wireless edge systems. In *Practice and Experience in Advanced Research Computing*, pages 1–9. 2022.
[4] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart. Marble: Multi-agent reinforcement learning at the edge for digital agriculture. In *Proceedings of the 7th ACM/IEEE Symposium on Edge Computing*, 2022.
[5] J. G. Boubin, N. T. Babu, C. Stewart, J. Chumley, and S. Zhang. Managing edge resources for fully autonomous aerial systems. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 74–87. ACM, 2019.
[6] P. Czyżżak and A. Jaszkiewicz. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of multi-criteria decision analysis*, 7(1):34–47, 1998.
[7] K. Keahey, J. Anderson, M. Sherman, C. Hammock, Z. Zhen, J. Tillotson, T. Bargo, L. Long, T. Ul Islam, S. Babu, et al. Chi-in-a-box: Reducing operational costs of research testbeds. In *Practice and Experience in Advanced Research Computing*, pages 1–8. 2022.
[8] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock, et al. Lessons learned from the chameleon testbed. In *2020 USENIX annual technical conference (USENIX ATC 20)*, pages 219–233, 2020.
[9] C. Qu, J. Boubin, D. Gafurov, J. Zhou, N. Aloysius, h. Nguyen, and P. Calyam. Uav swarms in smart agriculture: Experiences and opportunities. In *2022 IEEE International Conference on eScience*. IEEE, 2022.
[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.