Adaptive ML-Driven Resource Management for Diverse Cloud Workloads

Ioannis Roumpos
IMDEA Software Institute
Universidad Politécnica de Madrid
giannis.roumpos@imdea.org

Thaleia Dimitra Doudali
IMDEA Software Institute
Madrid, Spain
thaleia.doudali@imdea.org

Abstract

Cloud platforms host diverse workloads under strict performance objectives while sharing limited infrastructure resources. To handle fluctuating demand, systems use mechanisms such as autoscaling and overcommitment to dynamically adjust resource allocations, which requires anticipating how resource usage will evolve over time. When these adjustments rely solely on historical usage, systems operate reactively. While this approach is simple and widely deployed, it often misses sudden demand spikes, leading to throttling and performance degradation. Instead, proactive systems forecast future demand and provision resources ahead of time, but their effectiveness is limited because cloud workloads exhibit highly diverse and unpredictable behaviors. This creates a key design challenge: a single forecasting model rarely generalizes across workloads, while per-workload specialization is too costly to sustain at cloud scale. This thesis tackles this challenge by seeking a balance between universal and per-workload modeling to enable practical, ML-driven resource management. The goal is to deliver high prediction accuracy with low overhead and build resource management mechanisms that effectively anticipate spikes, eliminate throttling, and reduce resource waste across highly diverse cloud workloads.

1 Introduction

Cloud infrastructures today support millions of virtual machines, managed databases, and serverless applications, each exhibiting time varying and unpredictable resource demands. Efficient resource management is essential to satisfy performance objectives and maintain high utilization. To achieve this, cloud systems employ a combination of mechanisms including overprovisioning, autoscaling, and scheduling to continuously adjust resource allocations in response to workload dynamics [3, 5, 16, 18, 22, 24, 28–30]. As an example, vertical autoscaling dynamically adjusts CPU and memory allocations of running services to better match actual demand with minimal disruption [2, 7, 19, 21, 23]. Existing production solutions [1, 23] follow *reactive* strategies, resizing resources based on observed utilization signals. While simple and easily deployable, such approaches can lag behind abrupt demand spikes, leading to throttling and performance degradation. Recent *proactive* alternatives [7, 17, 19, 21, 24] incorporate demand forecasting to anticipate workload changes and provision resources ahead of time for specific classes of cloud workloads, such as serverless functions [17] or jobs [24].

Cloud workloads are highly heterogeneous, spanning from VM-level traces, large-scale cluster jobs, serverless functions, and emerging agentic services, with substantial variability in their temporal behavior [9–11, 13, 14, 17, 19, 20, 35]. As a result, a single predictive model rarely performs well across all settings. At the other extreme, fully specialized per workload modeling, while feasible through AutoML [25, 27] frameworks, incurs significant operational and computational overhead at cloud scale. This exposes a core challenge: neither a single universal model nor a fully specialized per workload modeling provides a satisfactory solution in practice. The former struggles to capture workload diversity, and the latter is costly to deploy and maintain.

The *key question* that this thesis aims to address is how to balance generality and specialization while remaining effective and deployable. Different cloud use cases, including VM provisioning, serverless scaling, and agentic workloads, exhibit distinct characteristics and objectives, requiring adaptive yet scalable solutions. This thesis aims to build system-level resource management solutions, such as autoscaling, that integrate machine learning in a practical and adaptive way across diverse cloud workloads.

2 Overview

This thesis tackles the challenge of integrating machine learning into cloud resource management while maintaining both effectiveness and deployability. First, we characterize the behavior of existing prediction-based resource management mechanisms across a wide range of real workloads, examining how different temporal patterns (periodic, bursty, and irregular) shape their decisions. This reveals which generic temporal characteristics most strongly influence prediction accuracy and thus resource and performance efficiency. Second, we aim to perform an extensive evaluation of diverse predictors, including statistical models, machine-learning methods, deep learning, and emerging foundation models, to understand which classes of workloads each model handles best. Third, informed by these insights, we plan to design a curated model pool that enables selecting the most suitable model for each *workload use case* (VM, serverless, agentic) and *temporal pattern*, achieving a practical middle ground

between a single global model and full per-workload specialization. Finally, we will integrate this ML-driven selection mechanism into system-level resource-management policies, such as autoscaling, and evaluate it against state-of-the-art solutions, measuring improvements in resource efficiency and the reduction in throttling and performance degradation.

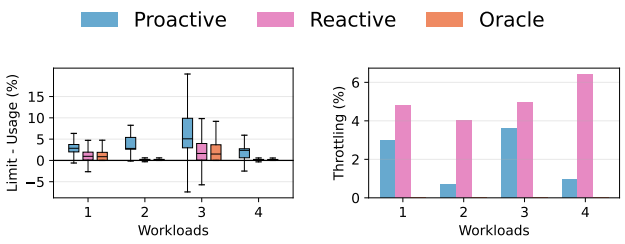
3 Preliminary Results

As a first step, this thesis focuses on VM-level vertical autoscaling and performas an experimental evaluation of the following three policies: (i) a *reactive* production-level autoscaler ([2]), (ii) a lightweight *proactive* policy based on simple historical forecasting techniques commonly adopted in practice [19–21, 24], and (iii) an *oracular* policy with perfect future knowledge of resource demand. We evaluate these policies across four representative workload types derived from production traces: (1) stationary workload with stable demand, (2) seasonal workload with periodic patterns, (3) spiky workload with short abrupt bursts, and an (4) irregular workload with highly variable and non-stationary behavior. Figure 1a shows the distribution of resource slack (allocated but unused CPU) versus insufficient CPU and 1b the percentage of throttling events (times where CPU was insufficient), across the afforementioned workloads with different temporal patterns.

We observe a consistent trade-off across workload types. The reactive policy exhibits the highest slack, reflecting conservative provisioning to mitigate performance risk. The predictive policy reduces slack, demonstrating the potential of forecasting, but increases throttling, especially for spiky and irregular workloads where abrupt demand changes are difficult to anticipate. The oracle policy achieves both minimal slack and zero throttling, establishing an upper bound on attainable performance. These results expose a clear gap between production behavior and achievable efficiency. They motivate the need for predictive autoscaling mechanisms that improve workload anticipation, while eliminating excessive throttling. At the same time, per workload model specialization is impractical at scale. This motivates the central direction of this thesis: designing practical, workload-aware model selection mechanisms that improve prediction quality without incurring the overhead of fully customized per workload pipelines.

4 Work To Be Done

Building on the preliminary results of Section 3, the next step is to bridge the gap between *ineffective reactive* autoscaling and *widely used lightweight proactive* ones. Current proactive autoscalers [19, 21, 24, 31] typically apply a single forecasting model across workloads (VMs in our use case), which limits performance under heterogeneous and bursty demand.



(a) Slack vs. insufficient CPU **(b)** Throttling events

Figure 1. Comparison of Proactive, Reaactive and Oracle autoscaling solutions across representative workloads.

Rather than fully specializing per workload, which is operationally costly, we will design a proactive vertical autoscaling framework that maintains a controlled pool of models, which will include statistical, machine learning and potentially deep learning ones. This design will enable model specialization at a coarser granularity and allow for its benefits in a practical way. Workloads will be grouped into clusters based on structural characteristics, enabling adaptive model selection while keeping ML overhead minimal. The objective is to minimize resource waste and throttling by aligning model selection for resource usage forecasting with workload behavior and cloud use case.

Beyond VM level autoscaling, the thesis will extend this approach to short lived and bursty serverless workloads and emerging agentic systems with interactive demand patterns. We will also study how workload-aware model selection can inform other mechanisms, such as scheduling and overcommitment. The broader aim is to develop ML-driven resource management strategies that remain practical, low overhead, and robust across diverse cloud workloads.

5 Related Work

There has been an extensive collection of prior work across distinct cloud use cases, like microservices [8, 12, 15, 32] and serverless functions [4, 6, 26, 33, 34]. Focusing on autoscaling, the adoption of ML-based solutions in production remains limited due to performance overheads, interpretability concerns, and engineering complexity [35]. Consequently, many systems rely on lightweight ML predictors [7] or simple statistical models [19–21, 24], which have proven effective for server-level resource forecasting [9]. Recent studies of serverless scaling show similar behavior: individual ML models rarely outperform seasonal statistical baselines because of the diversity of workload patterns [11, 14, 17], while ensemble methods can improve accuracy for spiky and irregular traces [11]. Therefore, relying on a single forecasting method is fundamentally limiting, strengthening the case for mechanisms that adapt model choice to workload behavior.

References

[1] 2026. Kubernetes. https://kubernetes.io/.

[2] 2026. Vertical Pod Autoscaler. https://kubernetes.io/docs/concepts/workloads/autoscaling/vertical-pod-autoscale/

[3] Noman Bashir, Nan Deng, Krzysztof Rzadca, David Irwin, Sree Kodak, and Rohit Jnagal. 2021. Take it to the limit: peak prediction-driven resource overcommitment in datacenters. In *Proceedings of the Sixteenth European Conference on Computer Systems* (Online Event, United Kingdom) *(EuroSys '21)*. Association for Computing Machinery, New York, NY, USA, 556–573. https://doi.org/10.1145/3447786.3456259

[4] Anirban Bhattacharjee, Ajay Dev Chhokra, Zhuangwei Kang, Hongyang Sun, Aniruddha Gokhale, and Gabor Karsai. 2019. BARISTA: Efficient and Scalable Serverless Serving System for Deep Learning Prediction Services. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*. 23–33. https://doi.org/10.1109/IC2E.2019.00-10

[5] Eric Boutin, Jaliya Ekanayake, Wei Lin, Bing Shi, Jingren Zhou, Zhengping Qian, Ming Wu, and Lidong Zhou. 2014. Apollo: Scalable and Coordinated Scheduling for Cloud-Scale Computing. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*. USENIX Association, Broomfield, CO, 285–300. https://www.usenix.org/conference/osdi14/technical-sessions/presentation/boutin

[6] Xinquan Cai, Qianlong Sang, Chuang Hu, Yili Gong, Kun Suo, Xiaobo Zhou, and Dazhao Cheng. 2024. Incendio: Priority-Based Scheduling for Alleviating Cold Start in Serverless Computing. *IEEE Trans. Comput.* 73, 7 (2024), 1780–1794. https://doi.org/10.1109/TC.2024.3386063

[7] Zihao Chen, Jiazhi Jiang, Jiangang Liu, Chao Zhang, Yuqi Diao, Yang Li, Hanmei Luo, and Peng Chen. 2025. Oceanus: Enable SLO-Aware Vertical Autoscaling for Cloud-Native Streaming Services in Tencent. In *Companion of the 2025 International Conference on Management of Data*. 323–335.

[8] Ka-Ho Chow, Umesh Deshpande, Sangeetha Seshadri, and Ling Liu. 2022. DeepRest: deep resource estimation for interactive microservices. In *Proceedings of the Seventeenth European Conference on Computer Systems* (Rennes, France) *(EuroSys '22)*. Association for Computing Machinery, New York, NY, USA, 181–198. https://doi.org/10.1145/3492321.3519564

[9] Georgia Christofidi, Konstantinos Papaioannou, and Thaleia Dimitra Doudali. 2023. Is Machine Learning Necessary for Cloud Resource Usage Forecasting?. In *Proceedings of the 2023 ACM Symposium on Cloud Computing* (Santa Cruz, CA, USA) *(SoCC '23)*. Association for Computing Machinery, New York, NY, USA, 544–554. https://doi.org/10.1145/3620678.3624790

[10] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. 2017. Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles* (Shanghai, China) *(SOSP '17)*. Association for Computing Machinery, New York, NY, USA, 153–167. https://doi.org/10.1145/3132747.3132772

[11] Yanlei Diao, Dominik Horn, Andreas Kipf, Oleksandr Shchur, Ines Benito, Wenjian Dong, Davide Pagano, Pascal Pfeil, Vikram Nathan, Balakrishnan Narayanaswamy, et al. 2024. Forecasting algorithms for intelligent resource scaling: An experimental analysis. In *Proceedings of the 2024 ACM Symposium on Cloud Computing*. 126–143.

[12] Yu Gan, Mingyu Liang, Sundar Dev, David Lo, and Christina Delimitrou. 2021. Sage: practical and scalable ML-driven performance debugging in microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Virtual, USA) *(ASPLOS '21)*. Association for Computing Machinery, New York, NY, USA, 135–151. https://doi.org/10.1145/3445814.3446700

[13] Jing Guo, Zihao Chang, Sa Wang, Haiyang Ding, Yihui Feng, Liang Mao, and Yungang Bao. 2019. Who Limits the Resource Efficiency of My Datacenter: An Analysis of Alibaba Datacenter Traces. In *Proceedings of the International Symposium on Quality of Service* (Phoenix, Arizona) *(IWQoS '19)*. Association for Computing Machinery, New York, NY, USA, Article 39, 10 pages. https://doi.org/10.1145/3326285.3329074

[14] Artjom Joosen, Ahmed Hassan, Martin Asenov, Rajkarn Singh, Luke Darlow, Jianfeng Wang, and Adam Barker. 2023. How Does It Function? Characterizing Long-term Trends in Production Serverless Workloads. In *Proceedings of the 2023 ACM Symposium on Cloud Computing* (Santa Cruz, CA, USA) *(SoCC '23)*. Association for Computing Machinery, New York, NY, USA, 443–458. https://doi.org/10.1145/3620678.3624783

[15] Shutian Luo, Huanle Xu, Kejiang Ye, Guoyao Xu, Liping Zhang, Jian He, Guodong Yang, and Chengzhong Xu. 2022. Erms: Efficient Resource Management for Shared Microservices with SLA Guarantees. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 62–77. https://doi.org/10.1145/3567955.3567964

[16] Shutian Luo, Huanle Xu, Kejiang Ye, Guoyao Xu, Liping Zhang, Guodong Yang, and Chengzhong Xu. 2022. The Power of Prediction: Microservice Auto Scaling via Workload Learning. In *Proceedings of the 13th ACM Symposium on Cloud Computing (SoCC '22)*. 355–369. https://doi.org/10.1145/3542929.3563477

[17] Nima Nasiri, Nalin Munshi, Marius Pirvu, Simon Moser, Vijay Sundaresan, Daryl Maier, Thatta Premnath, Norman Böwing, Sathish Gopalakrishnan, and Mohammad Shahrad. 2026. In-Production Characterization of an Open Source Serverless Platform and New Scaling Strategies. In *Proceedings of the 2026 European Conference on Computer Systems (EuroSys '26)*. ACM, Edinburgh, Scotland, UK. https://doi.org/10.1145/3767295.3769377

[18] Hiep Nguyen, Zhiming Shen, Xiaohui Gu, Sethuraman Subbiah, and John Wilkes. 2013. AGILE: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service. In *10th International Conference on Autonomic Computing (ICAC '13)*. 69–82. https://www.usenix.org/conference/icac13/technical-sessions/presentation/nguyen

[19] Anna Pavlenko, Joyce Cahoon, Yiwen Zhu, Brian Kroth, Michael Nelson, Andrew Carter, David Liao, Travis Wright, Jesús Camacho-Rodríguez, and Karla Saur. 2024. Vertically Autoscaling Monolithic Applications with CaaSPER: Scalable Container-as-a-Service Performance Enhanced Resizing Algorithm for the Cloud. In *SIGMOD*. https://www.microsoft.com/en-us/research/publication/caasper-vertical-autoscaling/ to appear.

[20] Olga Poppe, Tayo Amuneke, Dalitso Banda, Aritra De, Ari Green, Manon Knoertzer, Ehi Nosakhare, Karthik Rajendran, Deepak Shankargouda, Meina Wang, Alan Au, Carlo Curino, Qun Guo, Alekh Jindal, Ajay Kalhan, Morgan Oslake, Sonia Parchani, Vijay Ramani, Raj Sellappan, Saikat Sen, Sheetal Shrotri, Soundararajan Srinivasan, Ping Xia, Shize Xu, Alicia Yang, and Yiwen Zhu. 2020. Seagull: an infrastructure for load prediction and optimized resource allocation. *Proc. VLDB Endow.* 14, 2 (Oct. 2020), 154–162. https://doi.org/10.14778/3425879.3425886

[21] Olga Poppe, Qun Guo, Willis Lang, Pankaj Arora, Morgan Oslake, Shize Xu, and Ajay Kalhan. 2022. Moneyball: proactive auto-scaling in Microsoft Azure SQL database serverless. *Proc. VLDB Endow.* 15, 6 (Feb. 2022), 1279–1287. https://doi.org/10.14778/3514061.3514073

[22] Haoran Qiu, Subho S. Banerjee, Saurabh Jha, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. 2020. FIRM: An Intelligent Fine-grained Resource Management Framework for SLO-Oriented Microservices. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. USENIX Association, 805–825. https://www.usenix.org/conference/osdi20/presentation/qiu

[23] Red Hat. 2021. Vertical Pod Autoscaler in OpenShift Container Platform. https://docs.redhat.com/en/documentation/openshift_container_platform/4.9/html/nodes/working-with-pods#nodes-

pods-vpa. Accessed: 2026-02-05.

[24] Krzysztof Rzadca, Pawel Findeisen, Jacek Swiderski, Przemyslaw Zych, Przemyslaw Broniek, Jarek Kusmierek, Pawel Nowak, Beata Strack, Piotr Witusowski, Steven Hand, and John Wilkes. 2020. Autopilot: workload autoscaling at Google. In *Proceedings of the Fifteenth European Conference on Computer Systems* (Heraklion, Greece) *(EuroSys '20)*. Association for Computing Machinery, New York, NY, USA, Article 16, 16 pages. https://doi.org/10.1145/3342195.3387524

[25] Syed Yousaf Shah, Dhaval Patel, Long Vu, Xuan-Hong Dang, Bei Chen, Peter Kirchner, Horst Samulowitz, David Wood, Gregory Bramble, Wesley M. Gifford, Giridhar Ganapavarapu, Roman Vaculin, and Petros Zerfos. 2021. AutoAI-TS: AutoAI for Time Series Forecasting. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 2584–2596. https://doi.org/10.1145/3448016.3457557

[26] Mohammad Shahrad, Rodrigo Fonseca, Inigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, and Ricardo Bianchini. 2020. Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, 205–218. https://www.usenix.org/conference/atc20/presentation/shahrad

[27] Oleksandr Shchur, Caner Turkmen, Nick Erickson, Huibin Shen, Alexander Shirkov, Tony Hu, and Yuyang Wang. 2023. AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting. arXiv:2308.05566 [cs.LG] https://arxiv.org/abs/2308.05566

[28] Xiaoyang Sun, Chunming Hu, Renyu Yang, Peter Garraghan, Tianyu Wo, Jie Xu, Jianyong Zhu, and Chao Li. 2018. ROSE: Cluster Resource Scheduling via Speculative Over-Subscription. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 949–960. https://doi.org/10.1109/ICDCS.2018.00096

[29] Huangshi Tian, Suyi Li, Ao Wang, Wei Wang, Tianlong Wu, and Haoran Yang. 2022. Owl: Performance-Aware Scheduling for Resource-Efficient Function-as-a-Service Cloud. In *Proceedings of the 13th ACM Symposium on Cloud Computing (SoCC '22)*. 78–93. https://doi.org/10.1145/3542929.3563470

[30] Muhammad Tirmazi, Adam Barker, Nan Deng, Md E. Haque, Zhijing Gene Qin, Steven Hand, Mor Harchol-Balter, and John Wilkes. 2020. Borg: the next generation. In *Proceedings of the Fifteenth European Conference on Computer Systems* (Heraklion, Greece) *(EuroSys '20)*. Association for Computing Machinery, New York, NY, USA, Article 30, 14 pages. https://doi.org/10.1145/3342195.3387517

[31] Thomas Wang, Simone Ferlin, and Marco Chiesa. 2021. Predicting CPU usage for proactive autoscaling. In *Proceedings of the 1st Workshop on Machine Learning and Systems* (Online, United Kingdom) *(EuroMLSys '21)*. Association for Computing Machinery, New York, NY, USA, 31–38. https://doi.org/10.1145/3437984.3458831

[32] Yanqi Zhang, Weizhe Hua, Zhuangzhuang Zhou, G. Edward Suh, and Christina Delimitrou. 2021. Sinan: ML-based and QoS-aware resource management for cloud microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Virtual, USA) *(ASPLOS '21)*. Association for Computing Machinery, New York, NY, USA, 167–181. https://doi.org/10.1145/3445814.3446693

[33] Yuxuan Zhao and Alexandru Uta. 2022. Tiny Autoscalers for Tiny Workloads: Dynamic CPU Allocation for Serverless Functions. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 170–179. https://doi.org/10.1109/CCGrid54584.2022.00026

[34] Zhuangzhuang Zhou, Yanqi Zhang, and Christina Delimitrou. 2022. AQUATOPE: QoS-and-Uncertainty-Aware Resource Management for Multi-stage Serverless Workflows. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3567955.3567960

[35] Yiwen Zhu, Yuanyuan Tian, Joyce Cahoon, Subru Krishnan, Ankita Agarwal, Rana Alotaibi, Jesús Camacho-Rodríguez, Bibin A Chundatt, Andrew Chung, Niharika Dutta, Andrew Fogarty, Anja Gruenheid, Brandon Haynes, Matteo Interlandi, Minu Iyer, Nick Jurgens, Sumeet Khushalani, Brian Kroth, Manoj Kumar, Jyoti Leeka, Sergiy Matusevych, Minni Mittal, Andreas C. Müller, Kartheek Muthyala, Harsha Nagulapalli, Yoonjae Park, Hiren Patel, Anna Pavlenko, Olga Poppe, Santhosh Ravindran, Karla Saur, Rathijit Sen, Steve Suh, Arijit Tarafdar, Kunal Waghray, Demin Wang, Carlo Curino, and Raghu Ramakrishnan. 2023. Towards Building Autonomous Data Services on Azure. In *ACM SIGMOD*. https://www.microsoft.com/en-us/research/publication/towards-building-autonomous-data-services-on-azure/