Using Reinforcement Learning to Control Auto-Scaling of Distributed Applications

Gabriele Russo Russo
russo.russo@ing.uniroma2.it
University of Rome Tor Vergata
Rome, Italy

ABSTRACT

Modern distributed systems can benefit from the availability of large-scale and heterogeneous computing infrastructures. However, the complexity and dynamic nature of these environments also call for self-adaptation abilities, as guaranteeing efficient resource usage and acceptable service levels through static configurations is very difficult.

In this talk, we discuss a hierarchical auto-scaling approach for distributed applications, where application-level managers steer the overall process by supervising component-level adaptation managers. Following a bottom-up approach, we first discuss how to exploit model-free and model-based reinforcement learning to compute auto-scaling policies for each component. Then, we show how Bayesian optimization can be used to automatically configure the lower-level auto-scalers based on application-level objectives. As a case study, we consider distributed data stream processing applications, which process high-volume data flows in near real-time and cope with varying and unpredictable workloads.

ACM Reference Format:
Gabriele Russo Russo. 2023. Using Reinforcement Learning to Control Auto-Scaling of Distributed Applications. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion), April 15–19, 2023, Coimbra, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3578245.3585427

1 OVERVIEW

Nowadays distributed applications can largely benefit from the availability of large-scale and heterogeneous computing infrastructures, not limited any more to Cloud data centers. Such resource richness has enabled the development and adoption of an ever-growing number of data-intensive applications, which deal with large data volumes, often arriving at very high and unpredictable rates, and necessarily scale their execution over multiple processors and computing nodes. At the same time, because of the scale, dynamic nature and heterogeneity of modern infrastructures, it is increasingly difficult to identify suitable deployment and resource configurations for applications so as to guarantee the desired Quality-of-Service (QoS) while making efficient use of the computing resources. Furthermore, given the variability that often characterizes application workloads, configurations identified at development- or deployment-time are hardily effective in the long term and must be adjusted.

For these reasons, *self-adaptation* abilities are increasingly important for distributed applications, in order to autonomously respond to changes in their working conditions with minimal or no service degradation. For this purpose, both run-time adaptation *mechanisms* (i.e., the "knobs" available to alter one or more aspects of the controlled system) and *control policies*, which define when and how adaptation actions are triggered, must be identified.

In this talk, we focus on the problem of defining suitable auto-scaling policies. While this problem has been widely investigated for years in the field of Cloud computing [1], there are still some issues that make it challenging. In particular, we will consider the following key issues:

- *Model uncertainty.* Defining adaptation policies that rely on application performance models is increasingly challenging, because of the difficulty of obtaining accurate models of the highly variable workloads and the performance uncertainty introduced by heterogeneous and distributed infrastructures. For this reason, it is not surprising that the adoption of machine learning methods to drive system adaptation has been growing for years [5].
- *Adaptation overheads.* Adaptation has often a "cost" in terms of introduced overhead, especially when dealing with stateful applications, which require special care to always preserve state integrity. For instance, while adding or removing a replica of a web server is usually easy, auto-scaling a stateful service involves a specific reconfiguration process. As the introduced overhead may be significant, adaptation policies should take it into account.
- *Scalability issues.* Given the growing scale of both computing infrastructures and applications, centralized controllers can suffer from scalability issues, especially in highly distributed environments (e.g., Edge/Fog computing environments). Therefore, it is desirable to adopt fully (or, at least, partially) decentralized adaptation control architectures.

In this talk, we discuss an approach to control auto-scaling for distributed applications based on *reinforcement learning* (RL), which is a collection of learning methods for sequential decision-making [10]. RL allows agents (i.e., adaptation controllers) to cope with uncertainty about the underlying system by improving their decision policies based on experience collected at run-time. We present RL-based policies as part of a two-layered hierarchical adaptation framework, which overcomes the scalability issues of fully centralized solutions.

ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal

ACM ISBN 979-8-4007-0072-9/23/04.
https://doi.org/10.1145/3578245.3585427

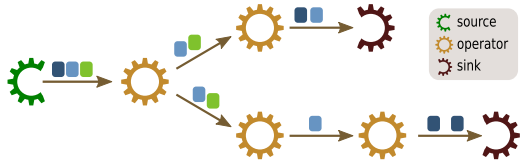

Figure 1: Example of DSP application.

Case Study: Data Stream Processing

As a case study for the talk, we consider distributed *data stream processing* (DSP) applications, which process high-volume data flows in near real-time [3] and cope with varying and unpredictable workloads and, thus, require effective auto-scaling abilities [2, 8]. Data *streams* are unbounded, ordered sequences of data, emitted by one or more sources. DSP applications are usually defined as directed acyclic graphs, whose vertices represent data sources and operators, and the edges represent streams flowing between them (see, Figure 1). *Operators* receive one or more streams as input, apply their processing logic (e.g., filtering, aggregation) and emit a new stream as the result, possibly updating an internal state. Data streams eventually reach *sink* vertices, which act as consumers of the produced results (e.g., dashboards, databases).

In practice, parallel replicas of each operator are executed across multiple – and possibly heterogeneous – processors and computing nodes to sustain higher data rates. By means of auto-scaling policies, the parallelism level and the type of computing resource in use for each operator can be modified at run-time depending on the workload. Specifically, scaling decisions aim to provision enough computational capacity to meet users' requirements (e.g., maximum processing latency), while avoiding resource wastage.

Auto-scaling is particularly challenging in the context of DSP, because parallelism reconfigurations come at the cost of significant overhead. Indeed, to preserve stream and state integrity, specific reconfiguration protocols must be adopted, slowing down or (more likely) interrupting normal data processing [6].

2 REINFORCEMENT LEARNING-BASED AUTO-SCALING

We tackle the problem of minimizing application resource usage while meetings users' Quality-of-Service (QoS) requirements concerning application performance and reconfiguration overheads. We discuss a hierarchical approach for auto-scaling where *application-level* managers steer the overall process by supervising *component-level* adaptation managers.

To cope with uncertainty about application performance, infrastructure conditions as well as workload dynamics, we exploit RL to determine local component-level policies. As such, the agents (i.e., the adaptation controllers) must choose suitable scaling decisions over time so as to minimize the long-term value of a cost function, which – in our formulation – accounts for performance, resource usage and scaling overhead. A key challenge with RL methods is the possibly long time required to learn a good policy. This is especially important when no historical information is available to train agents off-line and, instead, the whole training happens on-line. We tackle this challenge (i) exploiting function approximation techniques [9] and deep RL [7], which allow agents to learn over a reduced parameter space; and (ii) integrating partial model knowledge in the learning algorithm, which reduces the amount of information to learn at run-time.

To make sure that decentralized auto-scalers contribute to the satisfaction of application-level QoS requirements, an application controller must suitably tune the local cost function of each component auto-scaler (i.e., specifying a suitable local performance requirement and properly weighting the multiple objective terms). For this purpose, we discuss how black-box optimization techniques and, in particular, *Bayesian optimization* [4] can be exploited.

BIOGRAPHY

Gabriele Russo Russo is a PostDoc researcher at the Department of Civil Engineering and Computer Science Engineering of the University of Rome Tor Vergata, where he received his PhD degree in May 2021. His research interests span the area of distributed computing systems with emphasis on Quality-of-Service optimization and run-time self-adaptation. His current research mainly revolves around run-time resource allocation for data-intensive and serverless applications.

REFERENCES

[1] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah, and Philippe Merle. 2018. Elasticity in Cloud Computing: State of the Art and Research Challenges. 11, 2 (2018), 430–447. https://doi.org/10.1109/TSC.2017.2711009

[2] Valeria Cardellini, Francesco Lo Presti, Matteo Nardelli, and Gabriele Russo Russo. 2022. Run-Time Adaptation of Data Stream Processing Systems: The State of the Art. *ACM Comput. Surv.* 54 (2022), 36 pages. Issue 11s. https://doi.org/10.1145/3514496

[3] Marios Fragkoulis, Paris Carbone, Vasiliki Kalavri, and Asterios Katsifodimos. 2020. A Survey on the Evolution of Stream Processing Systems. *CoRR* abs/2008.00842 (2020). arXiv:2008.00842 https://arxiv.org/abs/2008.00842

[4] Peter I. Frazier. 2018. *A Tutorial on Bayesian Optimization.* Vol. abs/1807.02811. arXiv:1807.02811 http://arxiv.org/abs/1807.02811

[5] Omid Gheibi, Danny Weyns, and Federico Quin. 2021. Applying Machine Learning in Self-Adaptive Systems: A Systematic Literature Review. *ACM Transactions on Autonomous and Adaptive Systems* 15, 3, Article 9 (2021), 37 pages. https://doi.org/10.1145/3469440

[6] Thomas Heinze, Leonardo Aniello, Leonardo Querzoni, and Zbigniew Jerzak. 2014. Cloud-based Data Stream Processing. In *Proc. of 8th ACM Int'l Conf. on Distributed Event-Based Systems, DEBS '14.* 238–245. https://doi.org/10.1145/2611286.2611309

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, et al. 2015. Human-Level Control Through Deep Reinforcement Learning. *Nat.* 518, 7540 (2015), 529–533. https://doi.org/10.1038/nature14236

[8] Henriette Röger and Ruben Mayer. 2019. A Comprehensive Survey on Parallelization and Elasticity in Stream Processing. *ACM Comput. Surv.* 52, 2 (2019), 36:1–36:37. https://doi.org/10.1145/3303849

[9] Gabriele Russo Russo, Valeria Cardellini, and Francesco Lo Presti. 2019. Reinforcement Learning Based Policies for Elastic Stream Processing on Heterogeneous Resources. In *Proc. of 13th ACM Int'l Conf. on Distributed and Event-based Systems, DEBS '19.* 31–42. https://doi.org/10.1145/3328905.3329506

[10] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (2 ed.). MIT Press, Cambridge, MA, USA.