Causality-Based Service Adaptations for Elastic Computing Continuum Systems

Doctoral Candidate: Dipl.-Ing. Boris Sedlak
Advisor: Univ. Prof. Dr. Schahram Dustdar

Distributed Systems Group, TU Wien, Vienna, Austria.

Abstract. Computing Continuum (CC) systems are challenged to ensure the intricate requirements of multiple computational tiers. While some tiers, like the Cloud, can easily scale their processing resources to meet specific Service Level Objectives (SLOs), other tiers, like Edge computing, lack this abundance of resources and must adhere to alternative ways for elasticity. To that extent, creating a causal view into processing services allows to accurately infer the best possible way to adjust a service, and in further consequence, scale composite services distributed throughout the CC. We identify three challenges that must be tackled to achieve this: (1) to identify key factors that drive SLO fulfillment, it needs a transparent view into individual services and their precise implications to the processing hardware; given that, the next step is to (2) estimate the impact that processing services have on each other's SLOs, which is defined by the interactions and dependencies of services; lastly, (3) to ensure that services choose the best adaptations regardless of external perturbations, any causal model must be continuously adjusted according to new observations. This thesis proposes novel methods to achieve overarching SLO fulfillment in CC systems, which use Active Inference (AIF) to create a causal view of individual services as well as their interactions. Thus, the CC can be spanned with SLOs of different granularity, so that individual services, or compositions of services, can autonomously achieve an equilibrium.

1 Introduction

Distributed Computing Continuum Systems (DCCS), as envisioned in [1,6], are large-scale distributed systems composed of multiple computational tiers. Each tier serves a unique purpose, e.g., providing latency-sensitive services (i.e., Edge), or performant, scalable resources (i.e., Cloud). However, the requirements that each tier and the respective devices and services must fulfill are equally diverse. Assume that requirements would be evaluated in the cloud, e.g., by analyzing metrics and reconfiguring individual devices, massive amounts of data have to be transferred. Also, if edge devices fail to provide their service to a satisfying degree, the latency for detecting and resolving this is high.

Given the scale of the CC, requirements must be decentralized; this means that the logic to evaluate requirements must be transferred to the component that they concern. High-level requirements, i.e., Service Level Objectives (SLOs), must be disaggregated into smaller parts that can be ensured by the respective components. This would allow spanning the entire CC with SLOs so that each component contributes to high-level goals [2]. While it is one challenge to segregate SLOs, ensuring them is another. Requirements are versatile and may change over time, every component must itself discover how its SLOs are related to its actions. For this to happen, the device could use Machine Learning (ML) techniques to discover causal relations between its environment and SLO fulfillment [16]. This promotes the usage of Active Inference (AIF) [8], an emerging concept from neuroscience that describes how the brain continuously predicts and evaluates sensory information to model real-world processes. By extending individual CC components with AIF, they could develop a causal understanding of how to adjust their environment to ensure preferences (i.e., SLOs).

Ensuring SLOs autonomously (i.e., evaluating the environment to infer adaptations) makes components intelligent [11]; any system composed entirely of such intelligent, self-contained components becomes more resilient and reliable. No central logic must be employed to ensure SLOs; thus, higher-level components can rely on the SLO fulfillment of underlying components. While each tier has its characteristic SLOs, their tools for adaptation can have a different scale, e.g., fog nodes would be able to shift computations within clusters from devices that fail their SLOs. Such operations can consider environmental impacts (e.g., network issues) as well as heterogeneous device characteristics. The Cloud, as the next layer, would even have sweeping tools to ensure global SLOs.

To realize this vision, this thesis presents a causality-based framework for elastic service adaptations. Guided by AIF, individual services gradually develop a causal understanding of how to ensure their SLO. This includes a causal view into their internal dynamics and processing requirements; hence, it can be inferred under which circumstances they likely fulfill their SLOs. By using AIF, we ensure that causal models stay accurate and decisions are taken based on well-founded assumptions. In the next step, multiple causal service models can be merged into an overarching representation; this can depict service compositions, such as microservice pipelines. Thus, services can autonomously resolve SLO violations to achieve a wide-ranging equilibrium in the CC.

2 Problem Statement

The CC is designed to operate vast numbers of processing services on heterogeneous devices [6]; however, there is a gap for orchestration mechanisms that can evaluate and enforce SLOs through multiple CC tiers [14]. This thesis aims to fill this gap. In the following, the outlined problems are summarized in three concise research questions (RQs) that will guide the remaining proposal:

RQ.1 How to avoid sub-optimal service configurations and continuously assure the accuracy of causal service interpretation models?

To meet processing requirements (i.e., SLOs) at edge devices, it is possible to scale the quality of the output [5,9] as a form of elasticity – different service states, or configurations, thus define the service output. However, in real-world systems variable distributions can change at any point [12]; hence, any causal interpretation model used to find SLO-fulfilling configurations for individual services (**RQ.2**) or their compositions (**RQ.3**) must be continuously adjusted according to new observations. Thus, services could avoid sub-optimal configurations when the QoS is compromised due to temporary perturbations, confounding variables external to the model, or variable drifts.

RQ.2 How to model individual processing services and their implications to processing hardware so that they can autonomously ensure SLOs regardless of external perturbation?

When aiming to ensure requirements for the CC, the bottom-up approach is to start from the smallest unit that should be supervised: an individual processing service. Suppose this service gets constrained by a set of SLOs (e.g., response time), the question is whether the service will be able to fulfill these SLOs given its available resources. Hence, the type of hosting device has clear implications on the potential SLO fulfillment, e.g., a CV task might largely benefit from an available GPU. Given the amount of heterogeneous resources in the CC, estimating the potential SLO fulfillment of a device at a certain device type would allow optimizing the deployment at design time, or even during runtime.

RQ.3 How to model the interactions and dependencies between microservices deployed throughout the DCCS so that it becomes possible to estimate the impact they have on each other's SLOs?

Microservice architectures commonly form sequential architectures, where the output of one service becomes the input of another service [17]. While service compositions allow building more advanced solutions, interactions between services obfuscate the expected SLO fulfillment for individual microservices. Whenever dependent services can take a multitude of states, these can change the expected SLO fulfillment entirely. Hence, modeling causal implications and dependencies between devices allows advancing from one service (**RQ.2**) to understand the impact that they have on each other's SLOs.

3 Methodology

To answer the specified research questions, we propose a systems and operations research methodology based on experimental analysis and quantitative methods, extended with a preliminary literature review. As a result, the following three steps guide the methodology of this thesis:

1. Analyze the state-of-the-art; identify challenges and shortcomings
2. Conceptualize novel concepts and implement them as prototypes
3. Evaluate the artifacts empirically; compare them with the state-of-art

First, to analyze the state of the art, we conduct a literature review that captures the opportunities and challenges present in the CC; this particularly focuses on survey and vision papers. This supposedly provides an intuition on the current challenges, however, a clear picture of the shortcomings of existing approaches will only emerge when delving into the intersection with SLOs and elasticity. In particular, this must focus on the extent to which causal methods have been implemented – or can be implemented – to fulfill SLOs. As a result, we will obtain a list of shortcomings that must be addressed.

Second, to address these shortcomings, we aim to provide formal concepts with an adequate level of detail. This can revolve around mechanisms that extract causal dependencies from individual microservices or their compositions; the target maintains to optimize their SLO fulfillment. To underline the viability of any developed concept, we will develop an executable prototype that can be deployed and operated throughout the CC, i.e., on Edge devices and cloud servers alike. As a result, we obtain an artifact that can be shared and inspected by different researchers.

Third and lastly, we must compare the performance of the developed prototype(s) and compare it to state-of-the-art solutions; this should particularly focus on an operational overhead introduced by any management framework, and most prominently, any QoS or QoE metric that should be constrained by SLOs. In case the literature review does not render suitable baselines, the developed baselines should at least be compared to common heuristics, so that this thesis can serve as a baseline for future work.

Noteworthy, throughout the proposed research path, contributions will be made to the TEADAL project[1] – a European Union Horizon 2020 project for collaborative data transformation and exchange in federation-wide data lakes.

4 Related Work

To the best of our knowledge and given the references below, there exist no other works that aim to ensure SLO fulfillment in the CC through causal methods. However, there are numerous works that apply ML in Edge-Cloud environments,

[1] https://www.teadal.eu/

of which some aim to ensure SLOs: Chen et al. [3] present *CauseInfer*, which pinpoint root causes within a system through causal inference; for this, *CauseInfer* explicitly determines fault propagation paths. To mitigate SLO violations, Qiu et al. present *FIRM* [15] as a framework for fine-grained resource management; *FIRM* analyzes telemetry data to detect critical paths in microservice architectures. A similar approach is designed by Hao et al. [10], called *Nazar*, which uses mobile devices to find root causes in distributed systems. Although *FIRM*, *CauseInfer*, and *Nazar* report significant improvements for error mitigation during runtime, they neglect prescriptive actions during design time.

To guarantee performance SLOs through cloud-native resources, Nastic et al. presented SLOC [13]. However, their approach does not extend to edge devices, which is the case for Furst et al. [9], who evaluated elastic and non-elastic services at the edge for an image processing task under latency SLOs. *Octopus* [18], presented by Zhang et al., ensures SLO fulfillment during a CV task; for this, they used deep RL to provide the optimal device configuration for three parameters. Filinis et al. [7] discuss how SLOs can be ensured for serverless function chains in the CC. However, all these assume that functions can always be scaled by provisioning more resources, neglecting alternative ways to scale the quality.

SLOs are an efficient way to model and enforce requirements at the respective component. Nevertheless, the remaining question is whether components have the required scope to recover SLO failures (e.g., by offloading computation), but it is impractical to evaluate SLOs in the cloud (e.g., *MHP2P* [4]). Ad-hoc hierarchical structures could provide a remedy, which *Duan* [4] are the only ones to use among the related work. However, they all assume prior knowledge of which variables impact SLO fulfillment. Contrarily, our envisioned approach (1) gradually increases the SLO scope by forming device clusters that span the entire CC, and (2) evaluates causal relations among environmental variables to shift the load from impacted devices.

Acknowledgement

Funded by *European Union (TEADAL, 101070186)*

References

1. Beckman, P., et al.: Harnessing the computing continuum for programming our world. In: Fog Computing. John Wiley & Sons, Ltd (Apr 2020)
2. Casamayor-Pujol, V., Morichetta, A., Murturi, I., Donta, P.K., Dustdar, S.: Fundamental Research Challenges for Distributed Computing Continuum Systems. Information **14**, 198 (Mar 2023). `https://doi.org/10.3390/info14030198`
3. Chen, P., Qi, Y., Hou, D.: CauseInfer: Automated End-to-End Performance Diagnosis with Hierarchical Causality Graph in Cloud Environment. IEEE Transactions on Services Computing (2019)
4. Duan, Z., Tian, C., Zhang, N., Zhou, M., Yu, B., Wang, X., Guo, J., Wu, Y.: A novel load balancing scheme for mobile edge computing. Journal of Systems and Software **186**, 111195 (Apr 2022). `https://doi.org/10.1016/j.jss.2021.111195`

5. Dustdar, S., Guo, Y., Satzger, B., Truong, H.L.: Principles of Elastic Processes. Internet Computing, IEEE **15**, 66–71 (Nov 2011)
6. Dustdar, S., Pujol, V.C., Donta, P.K.: On Distributed Computing Continuum Systems. IEEE Transactions on Knowledge and Data Engineering **35**(4), 4092–4105 (Apr 2023). `https://doi.org/10.1109/TKDE.2022.3142856`
7. Filinis, N., Tzanettis, I., Spatharakis, D., Fotopoulou, E., Dimolitsas, I., Zafeiropoulos, A., Vassilakis, C., Papavassiliou, S.: Intent-driven orchestration of serverless applications in the computing continuum. Future Generation Computer Systems (May 2024). `https://doi.org/10.1016/j.future.2023.12.032`
8. Friston, K., Da Costa, L., Sajid, N., Heins, C., Ueltzhöffer, K., Pavliotis, G.A., Parr, T.: The free energy principle made simpler but not too simple (May 2023). `https://doi.org/10.48550/arXiv.2201.06387`
9. Fürst, J., Fadel Argerich, M., Cheng, B., Papageorgiou, A.: Elastic Services for Edge Computing. In: 2018 14th International Conference on Network and Service Management (CNSM). pp. 358–362 (Nov 2018)
10. Hao, W., Wang, Z., Hong, L., Li, L., Karayanni, N., Mao, C., Yang, J., Cidon, A.: Monitoring and Adapting ML Models on Mobile Devices (May 2023). `https://doi.org/10.48550/arXiv.2305.07772`
11. Kokkonen, H., Lovén, L., Motlagh, N.H., Kumar, A., Partala, J., Nguyen, T., Pujol, V.C., Kostakos, P., Leppänen, T., González-Gil, A., Sola, E., Angulo, I., Liyanage, M., Bennis, M., Tarkoma, S., Dustdar, S., Pirttikangas, S., Riekki, J.: Autonomy and Intelligence in the Computing Continuum: Challenges, Enablers, and Future Directions for Orchestration (Feb 2023). `https://doi.org/10.48550/arXiv.2205.01423`
12. Lu, S., Wu, J., Lu, P., Wang, N., Liu, H., Fang, J.: QoS-Aware Online Service Provisioning and Updating in Cost-Efficient Multi-Tenant Mobile Edge Computing. IEEE Services Computing (2023)
13. Nastic, S., Morichetta, A., Pusztai, T., Dustdar, S., Ding, X., Vij, D., Xiong, Y.: SLOC: Service Level Objectives for Next Generation Cloud Computing. IEEE Internet Computing **24**(3) (May 2020)
14. Pujol, V.C., Dustdar, S.: Towards a Prime Directive of SLOs. In: 2023 IEEE International Conference on Software Services Engineering (SSE). pp. 61–70 (Jul 2023). `https://doi.org/10.1109/SSE60056.2023.00019`
15. Qiu, H., Banerjee, S.S., Jha, S., Kalbarczyk, Z.T., Iyer, R.K.: FIRM: An Intelligent Fine-grained Resource Management Framework for SLO-Oriented Microservices. In: 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). pp. 805–825 (2020)
16. Sedlak, B., Pujol, V.C., Donta, P.K., Dustdar, S.: Designing Reconfigurable Intelligent Systems with Markov Blankets. In: Service-Oriented Computing. pp. 42–50. Springer Nature Switzerland (2023). `https://doi.org/10.1007/978-3-031-48421-6_4`
17. Velepucha, V., Flores, P.: A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges. IEEE Access (2023). `https://doi.org/10.1109/ACCESS.2023.3305687`
18. Zhang, Z., Zhao, Y., Liu, J.: Octopus: SLO-Aware Progressive Inference Serving via Deep Reinforcement Learning in Multi-tenant Edge Cluster. In: Service-Oriented Computing. Cham (2023). `https://doi.org/10.1007/978-3-031-48424-7_18`