Towards a methodology for building dynamic urgent applications on continuum computing platforms

Daniel Balouek-Thomert*, Eddy Caron†, Laurent Lefevre†, Manish Parashar*
{daniel.balouek, parashar}@utah.edu, eddy.caron@ens-lyon.fr, laurent.lefevre@inria.fr
*SCI Institute, University of Utah, UT, USA
†Inria, Univ Lyon, EnsL, UCBL, CNRS, LIP, 46 Allée d'Italie, 69007 Lyon, France

***Abstract*—Advanced cyberinfrastructure aims at making the use of streaming data a common practice in the scientific community. They offer an ecosystem that links data, compute, network, and users to deliver knowledge obtained from multiple data sources using large-scale computational models. However, integrating this heterogeneous data with time-sensitive systems is difficult due to a lack of programming abstractions that can allow data-driven reactive behaviors throughout the edge-to-cloud/HPC computing continuum. Here we present a methodology for incorporating contextual information into the application logic while taking into consideration the heterogeneity of the underlying platform and the unpredictability of the data. A fire science scenario that includes sensors at the network's edge for smoke detection and computational models launched in the cloud for wildfire simulation and air quality assessment serves as the inspiration for this method. We then discuss research directions for tackling similar scenarios with a particular focus on resource management and programming models.**

***Index Terms*—Urgent computing, Edge computing, Cyberinfrastructure, Uncertainty**

I. INTRODUCTION

Advanced Cyber Infrastructures (CIs) constitute major enablers for online analytics in science and engineering research. They provide holistic ecosystems comprising computing resources as well as data, software, networking along with coordination and user support. While integrating sophisticated data analytics for emerging applications deployed at the edge of the network presents additional architectural and technical challenges, CIs have a real potential to address today's global challenges, such as pandemic management [1] or wildfire management [2].

Urgent analytics describe data-driven application workflows that leverage distributed data sources to support critical decision-making [3]–[5]. They aim at identifying critical events, and accelerate responses under strict time constraints and stringent error thresholds. For example, recent work on Earthquake Early Warning (EEW) systems builds on geographically distributed infrastructure to integrate multiple types of sensors for increasing the accuracy of detection while ensuring an efficient computation in terms of response time and robustness to partial infrastructure failures [6], [7].

Offering programmable reactive behaviors for applied models under platform uncertainty, or seamlessly integrating contextual information into the application logic while overcoming platform heterogeneity and data unpredictability, is one of the grails of urgent analytics. These two features are constrained by a number of factors. First, it is impossible for the domain scientist to satisfy the demand for time constraints or utility measures on their own because the target infrastructure and context are unknown during the design phase of applied models. Second, the applied models must maintain a separation between the concerns of domain experts (familiar with the applications) and those of resource managers (familiar with the infrastructure) in order to retain a certain level of genericity throughout the platform. Third, it is crucial to take into account the shared nature of the sensors and resources in CIs when selecting resource or performing actuations.

Due to these limitations, it is required to investigate a novel, comprehensive technique that integrates end users, resources, and services to enhance computing on CIs deployed along the Edge-to-Cloud Computing Continuum. Although the literature discusses cloud-centric models for capacity, latency, and mistakes, current efforts have not taken into account situations where immediate analytics on CI are required [8]. This methodology was inspired by reports of Utah's air pollution as a result of the wildfires in California [9]. The SAGE platform, a cutting-edge CI aimed at AI-driven research at the edge[1], is equipped with sensors that could help classify, forecast, and connect these occurrences to better alerts and responses.

In this paper, we introduce a methodology for defining strategies for the deployment of urgent analytics on CI and continuum resources. The underlying premise of our strategy is that the interactions between the platform's different subsystems and its complexity may be reduced to an abstract state machine with guarantees between its various transitions. Urgent applications require system management to balance requirements and costs as well as programming support to react to unforeseen events. This work is driven by an AI-enabled fire science scenario with sensors distributed over California, Oregon, and Utah. The scenario is based on the use of numerous plugins for monitoring and responding to fire events across various states and a variety of data sources, including concentration of air pollution, cloud movements, and smoke detection from wildfires. This will make it easier to develop original and important applications for SAGE

[1]SAGE - Cyber Infrastructure for AI at the Edge - https://sagecontinuum.org/

and similar CIs and provide a better understanding of the requirements for urgent analytics.

This paper's contributions are presented in three sections:

- A case study on a fire science scenario in Section II
- A methodology for defining policies for urgent analytics in Section III
- A research plan with challenges in programming models and resource management in Section IV

II. Use case

In this section, we introduce the features of urgent analytics, and the fire science use case that motivates this work. We then discuss the requirements associated with executing urgent analytics across CI spanning the Edge-to-Cloud Computing Continuum.

A. Urgent Computing

Urgent Computing refers to a class of time-critical scientific applications that leverage distributed data sources to facilitate important decision-making in a timely manner. The overall goal of Urgent Computing's overarching objective is to anticipate scenarios' outcomes in time to avert or lessen their unfavorable outcomes. For instance, responding to a Distributed Denial-of-Service (DDoS) cyberattack on a network of geographically dispersed devices like ATMs requires gathering the local state of each device, turning it into a global understanding of the network, characterizing the observed phenomenon using an applied model, and then initiating the necessary actions [10].

Urgent analytics differ from other types of analytics because of the distribution and nature of the data, the complexity of the models, the strict error thresholds, and the time constraints. These workflows' execution must combine the demand for a significant amount of processing power to reduce errors with the requirement to make sure that crucial data streams are processed and assimilated promptly [3], [4].

B. Fire scenario

Wildfires are among the most destructive natural disasters, destroying homes, wildlife habitat and timber, and polluting the air with emissions harmful to human health. Among many others in recent times, California's Dixie Fire has burned closer to a thousand square miles in August 2021 [11]. The claims of increased air pollution in Utah as a direct effect of the wildfires in California served as the driver for our effort [9]. The motivating scenario is predicated on the utilization of a large number of applied models for the purpose of following and responding to fire events across a variety of states and different data sources. These plugins track wildfire events, track smoke plumes, and model downwind air pollution concentrations associated with wildfire smoke.

There is a significant possibility that these wildfire events might be recognized, predicted, and linked to better alerts and reactions using sensors spread over California, Oregon, and Utah[2]. Through a three-step approach, we aim to realize this possibility. First, the HPWREN network's field cameras[3] are situated in California as a collaborative, networked cyberinfrastructure. To extract photos and increase trust in detecting smoke events, camera feeds are processed using smoke detection models. Second, a Fire Simulator (FS) gathers latitude/longitude data from the cameras and produces fire metrics relating to the intensity of a starting fire. Finally, the Air Quality (AQ) Model runs a Lagrangian transport model with inputs from gridded weather datasets and fire emissions to produce a map of air pollution concentrations. The output is subsequently applied to decision-making or to initiate further sensing. Such a scenario's specification calls for expertise from several fields and interdisciplinary cooperation.

[2]https://www.anl.gov/article/new-sensing-platform-deployed-at-controlled-burn-site-could-help-prevent-forest-fires

[3]http://hpwren.ucsd.edu/cameras/

C. Leveraging Cyber Infrastructures

Training and inference of applied models for machine learning (ML) is one of the key vectors for exploiting resources across the computing continuum. [12], [13]. This paradigm offers notable improvements for urgent analytics that seek to pinpoint extreme events and speed up response by setting off the necessary responses. Data inference, for instance, can be carried out in real-time close to ATM machines in the case of a DDoS detection model, and only pertinent data is sent to the cloud. Additionally, intermediate data from many Edge devices can be combined to enable quick decision-making based on utility or location-based features. While such requirements make large CIs ideal enablers due to their holistic approach that leverages end-users, resources, and services at the logical extreme of the network and along the data path [14], [15], exploiting resources across the continuum is challenging for urgent analytics [5].

In order to provide urgent analytics, developers and service providers must coordinate data-driven operations while taking into consideration the extreme system heterogeneity and the unpredictability brought on by the availability of data. Thus, for the end-to-end performance of these applications spread across the continuum, the adaptability of the resources and computational channels between the edge and the cloud, as well as programming abstractions that react at runtime to unforeseen occurrences, are required. Even though these problems are interconnected, developers cannot fix them on their own because the target infrastructure is unknown at the time of urgent analytics design. This led to the demand for a comprehensive and consistent methodology to produce and offer urgent analytics in light of a CI ecosystem.

III. Methodology for building urgent analytics

The methodology presented in this Section is meant to serve as a general approach for developing urgent analytics across CIs. By bridging active learning and distributed observations in relation to the content of data, the cost of computations, and the urgency of results, this methodology target the delivery and composition of urgent analytics in an intelligent manner. This approach is inspired by a growing body of literature on multi-armed bandits [16].

In order to optimize the trade-off between resource costs and latency expectations, users are forced to make increasingly difficult decisions at runtime since user demand and infrastructure statuses are constantly changing and are challenging to predict with sufficient accuracy. The specification of reactive behaviors while delivering urgent analytics over the Edge-to-Cloud continuum involves all participants in the CI ecosystem to varied degrees.

The **subject-matter experts** in wildfire mitigation, such as firemen, have in-depth understanding of physical phenomena and how real-world actions affect the environment. The **domain scientist** seeks to understand how an event spreads and enhances models to incorporate on-field activities. The **infrastructure manager** is in charge of overseeing the network of distributed instruments and the processing resources required to put sensor-driven models into action. The **developer** then implements data-driven workflows that integrate the enabling infrastructure with the applied models. The fact that these actors share resources and there is uncertainty over the availability of data makes it more difficult for them to communicate.

The following case considers the management of a CI with long-running functions for transforming unstructured data into knowledge and on-demand analytics triggered by applied model outputs. We assume that in order to be supplied, a result on demand must adhere to strict time and quality requirements. The CI platform offers tools for establishing services and choosing computing resources so that data processing can be scaled up or down depending on the situation. While more powerful resources are available in the network's core, either as internal CI resources or as external cloud/HPC resources, sensors are situated alongside constrained computing resources at the network's edge.

The methodology's final working hypothesis is that feedback on expectations in terms of timing and result quality is easily accessible for each result or data product supplied via the CI. The remainder of the Section breaks down the key elements used to establish dynamic policies for managing urgent analytics.

A. Decisions

The first step is to identify how we are going to control the CI. The decisions represents discrete actions that are based on the state of the system at a certain moment. We argue it is important to represent decisions that are actually actionable by the system at any time during the execution.

Table I lists the decisions available in the motivating fire science use case grouped according to their targeted impact.

Data Collection	Collect video streams
	Collect smoke measurements
Data Processing	Adjusting quality from raw data
	Triggering additional sensing
	Triggering additional applied models
Scheduling	Adjusting location of processing
	Requesting a priority queue in the CI cloud
	Deployment of external cloud/HPC resources
Human intervention	Requesting CI operator
	Requesting first responders

TABLE I: Decisions

B. Metrics

When installing an urgent workflow, the second stage is to decide how we will compare two or more possible outputs. The metrics stand in for the objectives connected with carrying out an urgent workflow. In the context of the fire science use case, we focus on performance metrics associated to the time and quality of the output.

C. Uncertainty

The third step consists in identifying new information that becomes available over time. The uncertainty represents events or data that was unknown when prior decisions were taken.

In the context of the fire science use case, the uncertainty is twofold (Table II). First, the starting fire itself has several means of detection (temperature, smoke, human intervention, etc.) and the associated location impacts the density of sensors and computing resources in nearby proximity. Second, the shared nature of resources introduces some variability about the computing and networking resources available when a fire event is detected.

Fire event	Intensity
	Location
	Mean of detection
	Nearby sensors
Infrastructure	Computing resources availability
	Network resources availability

TABLE II: Uncertainty

D. Transitions

The fourth step is to model the evolution of the system and establish links between decisions and uncertain information. The transitions represent equations from a pre-decision state to a post-decision state following the processing of uncertain information. For many applications, these equations (or state transition models) are unknown but we assume they can be approximated using observations samples and refined over time.

E. Objective functions

The last step captures the metrics used to evaluate performance, and provides the basis for searching over policies that determines decisions. Particularly, the current methodology aims at writing objective functions in terms of decisions and variables associated with the system states. The description of the system state can be extensive by nature, however, our approach focuses on variables that are related to decisions of interest. For example, the location of a sensor constitutes relevant information for triggering additional resources in a given area. On the contrary, the battery level of the sensors does not participate to an actionnable decision. Even though this information would be available, it is not considered in the system state.

F. Proof of concept

The SAGE CI offers a substantial national infrastructure to support AI at the network edge. SAGE is an excellent example of the kind of testbed that provides a wide range of system variability for analyzing our recommended methodology. The tremendous unpredictability of data, resources, and services throughout the Edge/Cloud-HPC continuum amplifies this heterogeneity. The SAGE catalog of plugins proposes applied models produced regardless of the device or runtime scenario in which they will be employed[4]. These plugins' scheduling does not account for customized behaviour when data content includes time-sensitive instructions or extra calculations. Application formulations and programming abstractions that allow developers to reveal inherent flexibilities and trade-offs as well as create policies and procedures that could lead to runtime adjustments are needed to deal with this variability.

IV. Research challenges for deploying urgent applications

We highlight three research challenges in this Section that should be taken into account when using the methods outlined in Section III. First, models for comprehending the evolution of system-wide metrics are needed when writing transitions between system states (IV. D). Therefore, new software abstractions are needed in order to give CI applications the capacity to trade cost and quality (IV. F) using platform-aware plug-ins. Finally, in order to assure consistency of decisions, picking decisions (IV. A) would benefit from a distributed control layer.

A. Understanding the variability of the resources composing the continuum

The right comprehension of the performance of certain deployments of urgent analytics (mapping of the various services across the continuum) and the corresponding limits of the underlying infrastructure are early challenges that applications developers and resource managers must overcome. The goal of this research activity is to create a general model for predicting urgent analytics performance over the continuum. An analytical model that incorporates state variables that describe the system, decision variables that inform what is to be controlled, external information that influences the system, transitions that describe the system's evolution, and finally objective functions as a way to evaluate performance metrics is absolutely necessary. While literature on cloud-centric models for capacity, latency, and mistakes exists, recent works have not taken into account scenarios involving the desire for urgent analytics. In order to develop a predictive engine for assessing the availability and efficiency under various stressing load situations, this research effort will involve developing and validating a collection of models describing the dynamics of the continuum.

[4]https://portal.sagecontinuum.org/apps/explore

B. Delivering software abstractions to program urgent analytics

Data gathering, data filtering, data processing, and the delivery of data products are critical components of the lifecycle of an urgent analytics program. These steps must be tuned in real time based on the complexity of the calculations, cost of the data, and time sensitiveness of the output. In order to process all video streams from a sizable monitored region, for instance, a query must be made without knowing the precise number and positions of the cameras that are currently in operation. The requested data can be larger in size or contain more information than anticipated, which will cause a resource selection to choose the most suitable computational resources within the continuum. It is necessary to abstract and automate the process of finding data sources and choosing computational resources. Otherwise, it creates a barrier of entry for the implementation of urgent analytics by taking developers away from the application logic.

We propose to build this implementation based on approximation techniques for data collection [17], decoupled interaction for resource discovery [18], attribute-based profiles for microservices [19] and location aware routing of messages [20]. We aim at validating this activity by identifying qualitative improvements in design and organization (ease of building urgent analytics), while evaluating implementation and operation in quantitative aspects (improvement of the application performance). The outcome is the creation of repeatable processes and artifacts that will be used to develop a realistic wildfire scenario across distributed sites.

C. Enabling control of the continuum through model-driven feedback

The Computing Continuum presents new issues relating to its control and adaptation while aggregating the architectural and algorithmic challenges of its subcomponents. Subcomponents naturally impact one another when undertaking urgent analytics since they compete on the same playing field. Building a holistic perception of the system is difficult since it is difficult to foresee the demand for urgent analytics and infrastructure volatility with adequate accuracy. Specifically, use cases where edge devices operate with insufficient or partial knowledge of the infrastructure lead to decisions being made in isolation that might negatively affect the overall performance [21], [22]. More research is needed to examine appropriate abstractions with current technologies because seminal work on knowledge planes suggests that identifying appropriate high-level abstractions based on AI and cognitive tools (rather than algorithmic approaches) are best suited to meet adaptation objectives level. We suggest looking for a distributed control layer between the continuous infrastructure and urgent analytics applications. Our working hypothesis for this problem is that model-driven feedback can help handle the combined effects of the many concurrent autonomy-related tasks that will be active across the infrastructure.

In addition to the anticipated cost reductions and performance enhancements, overcoming these research obstacles will enable the introduction of unique context-aware adaptations. This will improve understanding of the conditions for decentralized analytics and make it simpler to design unique and significant applications for CIs and related dynamic data systems.

V. Related Work

Motivation: supporting the mapping and reconfiguration of urgent analytics onto massively distributed resources.

Configuration adaptation for Edge-based Systems: Analyzing concurrently multiple data streams with limited resources forces resource-quality trade-offs. As the incoming data are processed concurrently, resources available to each data stream are often unknown. Online/offline configurations adaptation is currently a promising solution to address the issue of limited resources [23]–[27]. In [26], Wang *et al.*adopt an offline configuration adaptation and bandwidth allocation strategies to address the issue of limited resources between IoT devices and edge nodes. Systems in [24], [27] adopt an online configuration adaptation algorithms for video analytics in Edge computing. However, these systems focus on analysis stages, but not on the complete workflow. Additionally, they only target Edge-based video analytics applications.

Deep Learning for self-adaptive systems: Deep Learning techniques have been applied to reduce large adaptation spaces of Self-Adaptive systems [28], [29]. DLASeR applies a deep learner first to reduce the adaptation space for the threshold goals and then ranks these options for the optimization goal [30]. In [31], authors perform an adaptation of the search space granularity to have a trade-off between precise decisions and performance, using a constructivist approach. Other approaches use plannings and re-use of sequences of actions to guide decisions [32]–[34]. While these works map control properties to software qualities in the face of uncertainty, they do not perform evaluations with large dimensions.

VI. Conclusion

The scientific drivers of this paper are the delivery and composition of applied models for AI-driven fire science and the minimization of the effects of wildfires across states. The implementation of such workflow is based on three principal areas of study: (1) the development of reusable models for capturing edge resource dynamics to understand the availability and effectiveness of CIs. In order to support the autonomous reconfiguration of applications, resources, and services under constraints, particularly for urgent applications, (2) we intended to explore software abstractions for programming reactive and time-critical analytics across the edge-cloud/HPC continuum, and finally (3) addressing cost/benefit tradeoffs at runtime in a cross-layer manner.

The results are not only anticipated to advanced the fields of data-driven analytics and resource management, but also to have broad impacts on repeatable artifacts and processes in order to encourage multidisciplinary teams of application specialists, end users, developers, and researchers to enable integrated solutions.

Acknowledgements

This research is supported in part by the NSF under grant number OAC 2238064. The research was conducted as part of the SCI Institute at the University of Utah.

References

[1] Marynia Kolak, Xun Li, Qinyun Lin, Ryan Wang, Moksha Menghaney, Stephanie Yang, and Vidal Anguiano Jr. The us covid atlas: A dynamic cyberinfrastructure surveillance system for interactive exploration of the pandemic. *Transactions in GIS*, 25(4):1741–1765, 2021.

[2] Ilkay Altintas. Building cyberinfrastructure for translational impact: The wifire example. *Journal of Computational Science*, 52:101210, 2021.

[3] Siew Hoon Leong and Dieter Kranzlmüller. Towards a general definition of urgent computing. *Procedia Computer Science*, 51:2337 – 2346, 2015. International Conference On Computational Science, ICCS 2015.

[4] Gordon Gibb, Rupert Nash, Nick Brown, and Bianca Prodan. The technologies required for fusing hpc and real-time data to support urgent computing. In *2019 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*, pages 24–34. IEEE, 2019.

[5] Daniel Balouek-Thomert, Ivan Rodero, and Manish Parashar. Harnessing the computing continuum for urgent science. *ACM SIGMETRICS Performance Evaluation Review*, 48(2):41–46, 2020.

[6] K. Fauvel, D. Daniel Balouek-Thomert, D. Melgar, P. Silva, A. Simonet, G. Antoniu, A. Costan, V. Masson, M. Parashar, I. Rodero, and A. Termier. A Distributed Multi-Sensor Machine Learning Approach to Earthquake Early Warning. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[7] Marco Esposito, Lorenzo Palma, Alberto Belli, Luisiana Sabbatini, and Paola Pierleoni. Recent advances in internet of things solutions for early warning systems: A review. *Sensors*, 22(6):2124, 2022.

[8] Daniel Balouek-Thomert, Ivan Rodero, and Manish Parashar. Evaluating policy-driven adaptation on the edge-to-cloud continuum. In *2021 IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*, pages 11–20. IEEE, 2021.

[9] Liji M David, AR Ravishankara, Steven J Brey, Emily V Fischer, John Volckens, and Sonia Kreidenweis. Could the exception become the rule¿uncontrollable'air pollution events in the us due to wildland fires. *Environmental Research Letters*, 16(3):034029, 2021.

[10] B Ramesh, J Vineeth Kumar, Vuppala Vyshnavi, and Mutha Prem. Artificial intelligence framework for cyber attack detection and notifying using machine learning techniques.

[11] Cal Fire, Shasta County, and Unified Command Fires. California statewide fire summary. *Ertugrul, M., Varol, T., Ozel, HB, Cetin, M. and Sevik, H*, pages 1–17, 2021.

[12] J. Chen and X. Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[13] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey, 07 2019.

[14] D. Balouek-Thomert, E. G. Renart, A. R. Zamani, A. Simonet, and M. Parashar. Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows. *The International Journal of High Performance Computing Applications*, 33(6):1159–1174, November 2019.

[15] P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck. Harnessing the computing continuum for programming our world, 2019.

[16] Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[17] Martin Hirzel, Scott Schneider, and Buğra Gedik. Spl: An extensible language for distributed stream processing. *ACM Trans. Program. Lang. Syst.*, 39(1), March 2017.

[18] Nanyan Jiang, Andres Quiroz, Cristina Schmidt, and Manish Parashar. Meteor: a middleware infrastructure for content-based decoupled interactions in pervasive grid environments. *Concurrency and Computation: Practice and Experience*, 20(12):1455–1484, 2008.

[19] Zeina Houmani, Daniel Balouek-Thomert, Eddy Caron, and Manish Parashar. Enhancing microservices architectures using data-driven service discovery and qos guarantees. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pages 290–299. IEEE, 2020.

[20] Eduard Gibert Renart, Daniel Balouek-Thomert, and Manish Parashar. An edge-based framework for enabling data-driven pipelines for iot systems. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 885–894. IEEE, 2019.

[21] Challenges Monitoring the Arctic Tundra: The Distributed Arctic Observatory Cyber-Physical System - IEEE Internet of Things, Feb 2022. [Online; accessed 24. Feb. 2022].

[22] Frederico Alvares de Oliveira, Thomas Ledoux, and Rémi Sharrock. A framework for the coordination of multiple autonomic managers in cloud environments. In *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*, pages 179–188, 2013.

[23] Peng Liu, Bozhao Qi, and Suman Banerjee. Edgeeye: An edge service framework for real-time intelligent video analytics. In *Proceedings of the 1st international workshop on edge systems, analytics and networking*, pages 1–6, 2018.

[24] Junchen Jiang and others. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 253–266, 2018.

[25] Haoyu Zhang and others. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 377–392, Boston, MA, March 2017. USENIX Association.

[26] C. Wang and others. Joint Configuration Adaptation and Bandwidth Allocation for Edge-based Real-time Video Analytics. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 257–266, 2020.

[27] Woo-Joong Kim and Chan-Hyun Youn. Lightweight Online Profiling-Based Configuration Adaptation for Video Analytics System in Edge Computing. *IEEE Access*, 8:116881–116899, 2020.

[28] Francesco Restuccia and Tommaso Melodia. DeepWiERL: Bringing Deep Reinforcement Learning to the Internet of Self-Adaptive Things. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 844–853, July 2020. ISSN: 2641-9874.

[29] Alexander Palm, Andreas Metzger, and Klaus Pohl. Online Reinforcement Learning for Self-adaptive Information Systems. In Schahram Dustdar, Eric Yu, Camille Salinesi, Dominique Rieu, and Vik Pant, editors, *Advanced Information Systems Engineering*, Lecture Notes in Computer Science, pages 169–184, Cham, 2020. Springer International Publishing.

[30] Jeroen Van Der Donckt, Danny Weyns, Federico Quin, Jonas Van Der Donckt, and Sam Michiels. Applying deep learning to reduce large adaptation spaces of self-adaptive systems with multiple types of goals. In *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '20, pages 20–30, New York, NY, USA, June 2020. Association for Computing Machinery.

[31] Maxime Guériau, Nicolás Cardozo, and Ivana Dusparic. Constructivist Approach to State Space Adaptation in Reinforcement Learning. In *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 52–61, June 2019. ISSN: 1949-3681.

[32] Ashutosh Pandey, Ivan Ruchkin, Bradley Schmerl, and David Garlan. Hybrid Planning Using Learning and Model Checking for Autonomous Systems. In *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 55–64, August 2020.

[33] Amoghavarsha Suresh, Gagan Somashekar, Anandh Varadarajan, Veerendra Ramesh Kakarla, Hima Upadhyay, and Anshul Gandhi. ENSURE: Efficient Scheduling and Autonomous Resource Management in Serverless Environments. In *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 1–10, August 2020.

[34] Cody Kinneer, Rijnard van Tonder, David Garlan, and Claire Le Goues. Building Reusable Repertoires for Stochastic Self-* Planners. In *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 222–231, August 2020.