Towards Cognitive Self-Management of IoT-Edge-Cloud Continuum based on User Intents

Hui Song
SINTEF Digital, Norway
hui.song@sintef.no

Ahmet Soylu
Oslo Metropolitan University, Norway
ahmet.soylu@oslomet.no

Dumitru Roman
SINTEF Digital, Norway
dumitru.roman@sintef.no

***Abstract*—Elasticity of the computing continuum with on-demand availability allows for automated provisioning and release of computing resources as needed; however, this self-management capability is severely limited due to the lack of knowledge on historical and timely resource utilisation and means for stakeholders to express their needs in a high-level manner. In this paper, we introduce and discuss a new concept – *intent-based cognitive continuum* for sustainable elasticity.**

***Index Terms*—Computing continuum, intent-based computing**

I. Introduction

Cloud computing provides elastic computing resources with on-demand availability, without the need of direct active management by users and application developers. However, the self-management capability of cloud, as key to on-demand availability, is far from satisfactory. The reason is twofold: (i) cloud management platforms lack the intelligence for effective resource optimisation and adaptation; and (ii) stakeholders lack the way to tell the cloud what exactly they need (and thus be "over-served"). The problem is exacerbated when organisations struggle to manage their assets across multiple regions and providers, in addition to having on-premise and edge devices – the lack of effective management blocks the transition from central cloud to the computing continuum.

In order to alleviate the aforementioned issues and to ensure a sustainable cloud elasticity, in this paper, we introduce a new concept, namely *intent-based cognitive continuum*[1]. The concept is based on (i) the use of Artificial Intelligence (AI) techniques, instead of following predefined rules and policies, to manage the resources based on the self-built knowledge learned from historical data and active exploration of the target continuum; (ii) a unified data layer to support AI on top of distributed and heterogeneous resources in the continuum, inspired by digital twins (DT); and (iii) intent-based human-AI interaction to enable stakeholders to describe how they intend their application to perform in the continuum, at a high-level and from a business perspective. Currently, outside the domain of continuum management, combining cognitive decision-making with high-level human involvement is an active research topic, with promising results such as explainable AI (XAI) [1] and intent-based chatbots [2]. Similar research for continuum management is still absent; among the existing edge/fog computing approaches [3], such level of management is not seen and in the many attempts using machine learning (ML) to manage edge applications [4], stakeholder interaction, trust or explanation are not considered. In networking, intent-based management approaches has been used to achieve a high-level control by network operators [5]. However, in the current Intent-Based Networking (IBN) solutions, intents are normally modelled as an abstraction of a set of pre-defined network operations, and therefore they lack the support of unpredictable situations. We introduce AI to remove the requirement of pre-defining operators for user intents, to support more complicated management scenarios in the continuum. Our contribution is summarised as follows: (i) a novel concept of intent-based cognitive management to achieve the next level self-management of the IoT-Edge-Cloud continuum and (ii) a conceptual architecture with key technical components for the implementation of intent-based cognitive management.

The rest of the paper is structured as follows. In Section II, we introduce an example scenario. Section III describes the proposed intent-based cognitive continuum concept and its architectural elements. Finally, we discuss the related work, summarise the paper, and provide an outlook in Section IV.

II. Example Scenario

The increased use of mobile devices is pushing a media content provider to embed user-generated footage in live broadcasting (e.g., football games) to enrich the entertaining experience. The contents need to go through a flow of processing services before being published, such as checking privacy issues. The processing application is currently running on the public cloud. As the audience content increases, this infrastructure is under considerable pressure. The content provider is planning to expand to the continuum, e.g., installing edge devices with GPUs (the edge-AI cluster) to its in-field broadcasting van, adding local servers in its premises, etc., so that some data processing services can be executed closer to the source, avoiding unnecessary traffic to the cloud.

The obstacle for moving to the continuum is how to manage the distributed, heterogeneous, and dynamic resources, so that the continuum is both elastic (when peak loads arrive, more instances of data processing services must be created, to ensure an acceptable latency) and sustainable (when the traffic is low, devices not heavily used should be in sleep mode, to reduce the energy consumption of the broadcasting van). The

[1] We use the terms "cloud", "continuum" and "computing continuum" interchangeably, as the extension from cloud to hybrid and edge is inevitable.

This is the author accepted version of an article published in
2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)
https://doi.org/10.1109/UCC56403.2022.00055

adaptation is complex, with trade-off between computation and communication, placement and migration of services, configuration of networks. Such complex runtime adaptation cannot rely solely on pre-defined management policies and calls for AI-powered adaptation. For example, supervised ML models can predict when the peak traffic may arrive based on historical data and make comprehensive plans for service placement, resource lifecyles, network configurations, etc.

The AI-based management decisions must comply with the intents of stakeholders. The content provider's in-field crew have the best understanding of a football game, and by "sensing" the atmosphere, they may intend either to seek for low latency making sure all contents are processed immediately, or to tolerate some possible delays. The AI should consider this intent when it decides whether to scale out microservices and activating edge devices to prepare for a predicted traffic increase, or to save some energy and cost. The in-field crew, normally with little IT background, need a way to express such intents to influence the AI's decision, and to understand and trust that the AI acts according to their intents.

III. Intent-based Cognitive Continuum

We propose a new concept, based on federated and hierarchical AI and higher-level intent-based human-cloud interaction, enabling sustainable elasticity and the self-management of the continuum – see Figure 1 for a high level architecture.

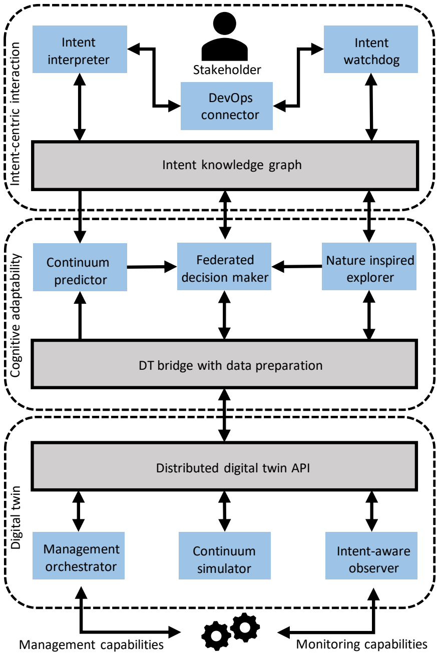

Fig. 1. High level architecture for intent-based human-cloud interaction.

A. Intent-based Human-Cloud Interaction

An intent captures how the stakeholders intend their application to perform on the target infrastructure, in a high abstraction level and from a business perspective. The intent-based continuum management ensures that the effect of automatic management complies with the intents of the stakeholders, and the stakeholders can trust the automated continuum management by understanding the rationale behind the decisions.

An ontology of possible and relevant intents within the scope of managing sustainable and elastic continuum captures different types of intents, such as constraints, preferences, expectations, goals, etc., the type of entities that are relevant to these intents, and the relationships among them. Under the ontology, all the intents and their entities form an ***intent knowledge graph***, stored in a distributed graph database accessible to the AI agents for querying.

Stakeholders explicitly specify their intents in semi-natural language. Developers may directly write: (a) "The latency should be stable", as part of the application's deployment specifications. An ***intent interpreter*** parses it into an intent of "performance requirement". Here, every such intent-claiming sentence must fall into one of the intent categories defined in the ontology, and that's why we call it semi-natural language. In addition to the explicitly claimed intents in a semi-natural language, implicit intents embedded in the stakeholder's actions are supported as well. For example, if the system operators suddenly add many devices into the edge-AI cluster in the field, it very likely means that they want to be more prepared to sudden traffic increase.

An ***intent watchdog*** collects and analyses: (i) the "subjective feedback" from AI agents on how they work with the intents; and (ii) the "objective feedback" directly from the DTs on whether the intents are fulfilled. The intent watchdog continuously monitors the intent KG with both subjective and objective runtime feedbacks to detect issues and possible causes. To achieve this, graph queries are used to identify relevant patterns, indicating situations that require stakeholder's attention, in the KG. In addition, continual ML is used to associate a series of changes in the graph, together with their contexts. The intent-based interaction, including the on-site interactive editing of intent claims, and the deep interaction of using intent to understand the AI work and revise original intents, is integrated into the DevOps practices of the stakeholders (i.e., ***DevOps connector***).

B. Digital Twins of the Managed Computing Continuum

The DTs of the continuum provide a unified data layer and a ***distributed digital twin API*** to facilitate both the intent layer and the AI agents. They cover a wide range of cloud management tasks; however, we focus on (i) microservice orchestration, (ii) microservice and container migration, (iii) device configuration and lifecycle, and (iv) network and data center management. The same digital twins concept can be applied on data pipelines [6]. A continuum comprises multiple subsystems, such as a resource group from a public cloud, a micro data center, a cluster of devices, etc. Each subsystem have its own DT agent, combining the management, simulation, and monitoring tools. Likewise, a group of subsystems has a DT agent that not only covers the global management and data, but also coordinates the agents of its subsystems. All the agents provide a standard API for reading (for monitoring and simulated data) and writing (for executing adaptation actions) on the DTs, together with meta-information.

The primary infrastructure and application management capabilities are already provided by the continuum through “management planes”, e.g., container orchestrator like Kubernetes supports the placement, replication, and migration of microservices that are wrapped up as containers. Software-Defined Networks allows the configuration of networks. Nevertheless, these capabilities are scattered in different planes, and without proper coordination with each other, the strategic management of all resources is prevented. A ***management orchestrator*** unifies these existing management planes and provides a standard interface to invoke the management actions. To achieve this, a taxonomy of management planes, capabilities, and the relationships between them are used. In addition to the runtime adaptation actions, the orchestrator also suggests long-term changes that needs the cloud administrator’s involvement, such as adding more devices into the cluster.

Cloud-native observability combines a comprehensive set of monitoring APIs, logging mechanisms, event recorders, etc., from different resources and technical stacks, to capture and deduce the real-time status of the infrastructure and the applications. There are two key challenges when extending observability to the continuum: (i) the resource constraint of edge devices means that one needs to be selective on what data can observed, and (ii) the complexity of the continuum means it is difficult to deduce the status from large amount of raw data. An ***intent-aware cloud observer*** uses the stakeholder’s intents, as well as the requirement from the AI agents, to configure what data should be monitored or logged, in which quality. A taxonomy links the intents and predictions of AI agents to concrete Key Performance Indicators (KPIs), an elastic device wrapper monitors a particular node and the applications on it, and an elastic aggregator gets the data from the wrappers and other aggregators and process them.

A distributed ***continuum simulator*** for the continuum, as part of the DT agent, for AI agents allows to explore and experiment management options. Simulator reduces the cost of performing the experiments in the real environment and allows for testing their performance where it would be unfeasible to use the real cloud. After identifying the nodes of the continuum that can be affected by the application and considering the requirements of the application (location, targeted users, etc.) and the predictions provided by the AI agents (possible deployment options, usage predictions, etc.), images of the simulated nodes are created from the historical data, considering the workload, the deployed applications, availability, etc. The network is emulated according to the real configuration, to connect the simulated nodes. The simulator also use the monitoring system of the continuum to evaluate the behaviour of the intent and the actions of the AI agents.

C. Intent-driven Cognitive Adaptability

For cognitive adaptability of complex continuums based on DTs and stakeholder intents, federated and hierarchical deep learning is exploited. Any subsystem in the continuum has an adaptation unit. Each unit obtains the raw data about the resources from the DT, and suggests adaptation actions by consulting the intents and collaborating with other units. The AI techniques in this part mainly focus on federated machine learning and evolutionary computing.

The AI needs high quality data about the resources and applications, and an effective way to execute the adaptation actions. DTs provide a standard interface to obtain data, but not all data are useful for all AI models; different AI methods require different data formats, such as a feature vector or time-series data; the data source and the AI agent that consumes the data are not always in the same place. A ***DT bridge*** tool shapes raw data to use for AI and assists AI designers in selecting the relevant data types, extracting these data from DT, preparing data into the needed format, and translating the adaptation decisions into changes in the DT. The core of this tool is based on identifying the features from the raw data and reducing its dimensionality to build efficient classifiers and predictors (i.e., representation learning).

A cognitive ***continuum predictor*** uses the current system metrics to predict the possible changes on key metrics, and the probability (confidence) of the changes. The tool covers too types of prediction: (i) the natural changes that may happen if the system keeps going as it is; (ii) the what-if changes, i.e., if a certain adaptation is executed, how some system metrics may change consequently. The high fluctuation of the system metrics represents a significant challenge, especially when the computing continuum status is the composition of multiple metrics that change dynamically. Another key characteristic is the variability of these measurements over time.

Reinforcement Learning (RL) techniques are used to look for the best adaptation actions for the actual and predicted system changes. A decision maker overlooking the entire continuum for multiple purposes is not feasible. Therefore, a novel federated learning architecture for distributed, multi-agent, and hierarchical decision making allows looking for the best adaptation actions for the actual and predicted changes (i.e.,***federated decision maker***). A subsystem in the continuum may have multiple decision agents for different purposes. A local supervisor, e.g., a RL agent, coordinates the adaptation solutions suggested by multiple agents, revise and combine them into the local adaptation actions. Similarly, the local decisions are reported to the supervisor of the parent system, which uses its global knowledge to revise the local decisions and coordinate them with decisions from other subsystems.

Using RL for adaptation tends to suggest adaptation actions having incremental impact on the current system configuration. Therefore, it may not be effective when facing large, systematic, and unexpected changes, such as introducing a new data center. Even without large changes, the legacy of previous adaptation actions may accumulate, causing ineffective adaptations. Bio-inspired AI techniques, such as evolutionary computing, are used to achieve the active exploration of the potential yet unconventional adaptation options. Under the hierarchical decision making structure, intents are considered as a special top-down information that are used as input for downstream decision making. A ***nature-inspired explorer*** plans for longer term and larger-scale system changes.

IV. Related Work, Summary, and Outlook

The management of the continuum is a very complex process, involving higher number of devices, of varying capabilities and features, with different objectives. The state-of-the-art approaches [7] require a wide external knowledge base, address single objectives, or focus on single management planes (e.g., approaches on service placement do not cover network adjustment). Simulation of the continuum has been used for developers to test deployments in the continuum, but it is not connected to real-time monitoring data [8]. System monitoring (requiring comprehensive solutions covering system instrumentation, data collection, aggregation, etc.), as the basis for management, is gaining interest [9]; however, the link from observability to automatic management is still weak.

Classical self-adaptive systems [10] rely on the provided knowledge, in the form of pre-defined scripts, policies, or alternative architectures, etc., to guide adaptation actions. Recent approaches [11] apply ML for the systems to build the knowledge themselves from historical data and relevant situations. As for human involvement, i.e., to guarantee that the adaptation complies with the expectation of the stakeholder, the existing approaches often use high-level goal models and requirement analysis methods [12]. However, the gap between goals and executable policies are still filled by software development. IBN [5] allows users to control the network through high-level intents, or purposes, from a business perspective. Commercial solutions, such as Cisco's IBN, still require a predefined action plan behind each of the intents they support. In general, IBN approaches usually fail to include considerations on more complex situations beyond connectivity and security policies, such as quality of service, resiliency, etc.

ML approaches have been used in continuum management [13], but mostly for single and isolated purposes, such as predicting future loads, optimising service locations, etc. Despite some attempts toward unified approaches based on specific AI techniques [14], what is still missing is the capability to create joint and coordinated learning and reasoning [15]. The complex architecture of the continuum also means the traditional single-model ML will not work in practice for real-time continuum management, and novel edge-native AI is needed [16]. Current AI-based cloud/continuum management approaches share the same bottleneck of AI application in many domains: the AI models incomprehensibly learn their own knowledge from data, leaving human stakeholders unable to interact, understand, or trust what the AI is doing. Human-AI interaction is widely considered in the design and development of AI-based systems in many domains [17]. Identifying the intents of human users from their inputs in natural or semi-nature languages is an advanced and effective way for such interaction, widely used in search engines, chatbots [18], etc. It is particularly useful when users are from multiple backgrounds, and when their needs from the system is complex and vague. In the other direction, XAI is gaining attention, aiming at the trustworthiness issues of using AI for critical tasks. The mainstream research is on tracing and visualisation of the ML models during training and inference [19].

We introduced the concept of intent-based cognitive continuum for embedding more intelligence in process of resource optimisation and adaptation on the computing continuum, while putting stakeholders centrally in the process. Since the concept is new, how it will actually be realised is an open issue. We plan to discuss various aspects and technology choices for all the architectural elements and develop a full proof-of-concept, in addition to identifying use cases from different domains for an overarching validation of the concept.

Acknowledgments. This research is partially funded by enRichMyData (HE 101070284), DataCloud (H2020 101016835), ERATOSTHENES (H2020 101020416), BigDataMine (NFR 309691), FLEET (NFR 309700), and SINTEF SEP-DataPipes.

References

[1] A. Barredo Arrieta *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[2] B. Luo *et al.*, "A critical review of state-of-the-art chatbot designs and applications," *WIREs Data Mining and Knowledge Discovery*, vol. 12, no. 1, p. e1434, 2022.

[3] P. Bellavista *et al.*, "A survey on fog computing for the internet of things," *Pervasive and Mobile Computing*, vol. 52, pp. 71–99, 2019.

[4] T. L. Duc *et al.*, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Computing Surveys*, vol. 52, no. 5, 2019.

[5] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *Proc. of the VTC2020-Spring*. IEEE, 2020, pp. 1–5.

[6] D. Roman *et al.*, "Big data pipelines on the computing continuum: Ecosystem and use cases overview," in *Proc. of the ISCC 2021*. IEEE, 2021, pp. 1–4.

[7] F. A. Salaht *et al.*, "An overview of service placement problem in fog and edge computing," *ACM Computing Surveys*, vol. 53, no. 3, 2020.

[8] R. Mahmud *et al.*, "ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software*, vol. 190, p. 111351, 2022.

[9] R. Picoreti *et al.*, "Multilevel observability in cloud orchestration," in *Proc. of the DASC/PiCom/DataCom/CyberSciTech 2018*. IEEE, 2018.

[10] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 2, 2009.

[11] T. R. D. Saputri and S.-W. Lee, "The application of machine learning in self-adaptive systems: A systematic literature review," *IEEE Access*, vol. 8, pp. 205 948–205 967, 2020.

[12] R. de Lemos *et al.*, *Software Engineering for Self-Adaptive Systems: A Second Research Roadmap*, ser. LNCS. Springer, 2013, pp. 1–32.

[13] Z. Zhong *et al.*, "Machine learning-based orchestration of containers: A taxonomy and future directions," *ACM Computing Surveys*, in press.

[14] C.-Z. Xu *et al.*, "Url: A unified reinforcement learning approach for autonomic cloud management," *Journal of Parallel and Distributed Computing*, vol. 72, no. 2, pp. 95–105, 2012.

[15] A. Morichetta *et al.*, "A roadmap on learning and reasoning for distributed computing continuum ecosystems," in *Proc. of the EDGE 2021*. IEEE, 2021, pp. 25–31.

[16] S. S. Gill *et al.*, "Ai for next generation computing: Emerging trends and future directions," *Internet of Things*, vol. 19, p. 100514, 2022.

[17] S. Amershi *et al.*, "Guidelines for human-ai interaction," in *Proc. of the CHI 2019*. ACM, 2019, p. 1–13.

[18] A. M. Rahman *et al.*, "Programming challenges of chatbot: Current and future prospective," in *Proc. of the R10-HTC 2017*. IEEE, 2017.

[19] S. Deo and N. S. Sontakke, "Usability, user comprehension, and perceptions of explanations for complex decision support systems in finance: A robo-advisory use case," *Computer*, vol. 54, no. 10, pp. 38–48, 2021.