2022 IEEE International Conference on Services Computing (SCC) | 978-1-6654-8146-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/SCC55611.2022.00060

Distributed Computing Continuum Systems

Schahram Dustdar
Distributed Systems Group
TU Wien
Wien, Austria
dustdar@dsg.tuwien.ac.at

***Abstract*—In this panel contribution, I will discuss my vision on the need of developing new managing technologies to harness distributed ``computing continuum'' systems. These systems are concurrently executed in multiple computing tiers: Cloud, Fog, Edge and IoT. This simple idea develops manifold challenges due to the inherent complexity inherited from the underlying infrastructures of these systems. This makes inappropriate the use of current methodologies for managing Internet distributed systems, which are based on the early systems that were based on client/server architectures and were completely specified by the application software.**

Keywords—Distributed Systems, Computing continuum, Edge Computing, Markov Blanket

I. Introduction

The development of software systems that perform on multiple computing tiers, including IoT, Edge, Fog, and Cloud, promises new opportunities for applications that require features provided by a specific tier. As we envision [1] increasingly complex applications, deriving features only from a specific computing layer is insufficient, demanding enlarging the perspective on all the computing tiers. This scenario entails a new paradigm, i.e., the aggregation of all computing tiers, also known as the computing continuum. One of the first issues that arise when dealing with applications requiring a computing continuum is how to manage them. The concurrent execution of an application on the entire computing continuum and its dependency on the underlying infrastructure makes it virtually impossible to specify its management solely on the application software. The methodologies developed for the Cloud tier, such as elasticity, do not adequately fit on the other tiers. Therefore, we aim at proposing a set of novel methodologies to manage distributed systems of the computing continuum [2].

II. Computng Continuum Requirements

The complexity and scale of these systems challenge current methodologies for managing distributed Internet-based systems, and methodologies such as the elasticity for Cloud systems and, most importantly, for online Cloud Computing, Fog Computing, and Edge computing are no longer fully applicable. Hence, every possible solution depends on the definition of novel, general, and adaptive requirements for computing continuum management (CCM) [2].

1) *Flexible representation*: Due to its scale and variable composition, CCM needs a flexible and adaptive representation of the system. Given the mutable state of the system architecture, its representation shifts as a variable feature of the system.

2) *Link with underlying infrastructure*: The characteristics of computing continuum systems depend on their underlying infrastructure and technologies tightly coupling with the application. The variety of IoT devices, Edge, Fog, and Cloud configurations compels any CCM methodology to consider the infrastructure as a key component.

3) *Causality relations*: We can see the computing continuum as an ecosystem; thus, a global perspective cannot consider modules and their action in isolation. Therefore, CCM must keep track of causal relationships between its components to understand the consequences of modifying a part or trace back any possible system issue.

4) *Temporal evolution*: Real-time systems in a computing continuum constantly evolve due to the external environmental transformation and the intrinsic differentiation and extensibility characteristics. Their state and structure can change with time; hence, taking this temporal evolution into account in CCM provides long-term adaptation capabilities tools. We can consider the system's variables as evolutionary, thus extracting environmental derivatives to understand the "direction" that a system is taking and act proportionally.

5) *Proactive adaptation*: The complexity of the ecosystem can lead to a cascade of failures as issues can propagate due to components' interconnections. In this regard, the CCM needs to act and adapt as soon as possible to overcome the system's agitations before they propagate.

6) *Learning framework*: The ecosystem complexity and scale make it impossible to draw a complete management plan in the design phase. Therefore, setting management methodologies inside a learning framework is required to provide incrementally better solutions and adaptations.
7) *Security and privacy embedded*: It is an increasing concern how these systems can expose security and privacy. Therefore, any management methodology has to have embedded means to tackle these types of issues.

[1] Dustdar S., Casamayor Pujol V., Donta P. K. (2022). On Distributed Computing Continuum Systems. *IEEE Transactions on Knowledge and Data Engineering, forthcoming*

[2] Morichetta A., Casamayor Pujol V., Dustdar S. (2021). A roadmap on learning and reasoning for distributed computing continuum ecosystems. IEEE International Conference on Edge Computing (EDGE 2021)

2474-2473/22/$31.00 ©2022 IEEE
DOI 10.1109/SCC55611.2022.00060