Distributing Time-Sensitive Applications on Edge Computing Environments

Eudald Sabaté¹,², Unai Perez¹,², Maria A. Serrano¹, Elli Kartsakli¹, and Eduardo Quiñones¹

¹Barcelona Supercomputing Center (BSC), Spain
²Technical University of Catalonia (UPC), Spain

Abstract—The proposed demonstration aims to showcase the capabilities of a task-based distributed programming framework for the execution of time-sensitive applications in edge computing scenarios, in the context of smart cities. Edge computing shifts the computation close to the data source, alleviating the pressure on the cloud and reducing application response times. However, the development and deployment of distributed applications is complex, due to the heterogeneous and dynamic nature of edge environments, where resources may not always be available.

To address these challenges, our demonstration employs COMPSs, a highly portable and infrastructure-agnostic programming model, that efficiently distributes time-sensitive applications across the compute continuum. We exhibit how COMPSs distributes the workload on different edge devices (e.g., NVIDIA GPUs or a Raspberry Pi), and how COMPSs re-adapts this distribution upon the availability (connection or disconnection) of devices.

DEMONSTRATION DESCRIPTION

Edge computing is a computing paradigm that shifts the computation close as possible to where the data is originated, alleviating the pressure on the cloud. This new paradigm, and the use of powerful embedded architectures, is becoming a very promising solution for large connected and distributed systems (e.g., those used in smart cities). However, the distribution and heterogeneity nature of the edge computing complicates the requirements analysis of time-sensitive applications.

We face this challenge using COMPSs, a software framework that aims to ease development and execution of applications for distributed infrastructures. COMPSs is composed of a task-based programming model and the runtime. Applications can be developed in Java, Python and C/C++, where the programmer identifies the functionalities to be executed as asynchronous operations (namely COMPSs tasks). Then, the runtime is in charge of distributing them in a transparent way, managing the parallelization, synchronization and data distribution aspects of these systems.

This demo illustrates the scheduling capabilities of COMPSs, that tries to minimize the overall response time of time-sensitive applications. Moreover, the COMPSs deployment and distribution processes are able to manage the dynamic availability of devices (appearing and disconnecting) in an edge computing infrastructure. As a result, the application workload is re-distributed, i.e., COMPSs is able to automatically react and re-adapt the mapping of COMPSs tasks to devices.

REFERENCES

