2016 IEEE/ACM Symposium on Edge Computing

Demo Abstract: RIAPS - A Resilient Information Architecture Platform for Edge Computing

William Emfinger*, Abhishek Dubey*, Peter Volgyesi*, Janos Sallai*, Gabor Karsai*
* Institute for Software-Integrated Systems, Vanderbilt University, Nashville, TN, USA

I. Overview

The emerging CPS/IoT ecosystem platforms such as Beaglebone Black, Raspberry Pi, Intel Edison and other edge devices such as SCALE [3], Paradrop[6] are providing new capabilities for data collection, analysis and processing at the *edge* [5] (also referred to as Fog Computing). This allows the dynamic composition of computing and communication networks that can be used to monitor and control the physical phenomena closer to the physical system. However, there are still a number of challenges that exist and must be resolved before we see wider applicability of these platforms for applications in safety-critical application domains such as Smart Grid and Traffic Control.

Two of the foremost challenges are related to resource uncertainty and network uncertainty. The resource uncertainty arises due to (a) the dynamism and heterogeneity of available resources, and (b) the increased likelihood of failure because the computing resources at the edge are not operating in a controlled environment, such as data centers. The network uncertainty is caused, e.g., by the shared wireless channels used at the edge that lead to varying network bandwidth. Additionally, security and resilience become important critical requirements and challenges at the edge, given that it is possible that multiple applications will share the computing and sensing resources, and, unlike in cloud computing, virtual machine- and container-based isolation are still not an easy option to adopt given the resource-constrained nature of the edge-computing nodes.

Solving these challenges require us to develop novel application programming paradigms that help developers write adaptive code to tolerate uncertainty, as well as to build a resilient application platform that not only provides core services like time synchronization, distributed data management and coordination, but also provides mechanisms to deploy and remotely manage the distributed applications that have a long lifecycle.

II. The RIAPS Platform

Our team is developing the core architecture, algorithms and programming paradigms for such a computing platform called RIAPS (Resilient Information Architecture Platform for Smart Grid) [2]. The software platform provides services for a wide variety of software applications where the services include (but are not limited to): distributed time management, planned and adaptive scheduling of computing and communication resources, a robust fault management, communication and coordination among concurrent activities, access to and control of physical system devices (i.e. sensors and actuators), and protection against and sustained operation of the system under security attacks.

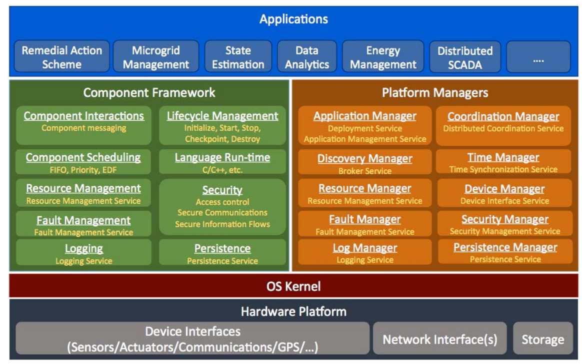

Fig. 1: A RIAPS Node

Developers will be enabled by the platform to implement adaptive intelligent applications that monitor, observe, control, and manage the physical system, while interacting with it at various physical locations, and on different levels of a sensing and control hierarchy.

Our immediate target for the platform is applications for the Smart Grid. However, as we will show in this demonstration, this architecture can also be used to design, deploy and manage applications for other domains as well. The demonstration described below will illustrate how the RIAPS platform can be used for implementing distributed, intelligent traffic light controllers, and how these can be tested on a Hardware-In-the-Loop (HIL) testbed.

III. Demonstration Scenario and the Test Bed

This section presents the traffic intersection control demo scenario and Cyber-Physical System (CPS) testbed used for testing and evaluation.

Testbed Architecture: The functions of a generic CPS testbed are 1) (re-)configurability with respect to the CPS, 2) execution of the CPS software on a realistic computing platform, 3) simulation or emulation of the communication network that connects the nodes, and 4) real-time simulation of the behavior of the sensors/actuators and the physical system. In addition to these concerns, the CPS testbed should behave similarly to the real system operating under fault conditions.

Because of these requirements, our CPS testbed was developed as a Hardware-in-the-Loop Emulation platform, where embedded computing nodes attached to emulated communi-

978-1-5090-3322-5/16 $31.00 © 2016 IEEE
DOI 10.1109/SEC.2016.23

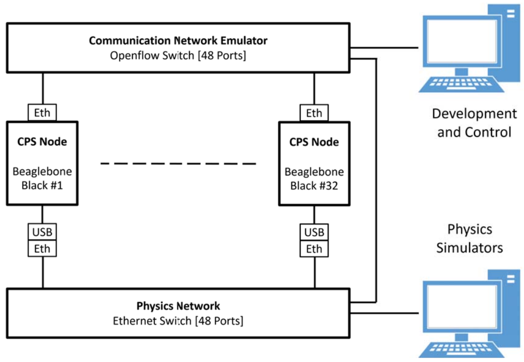

Fig. 2: Diagram showing the architecture of the CPS HIL testbed

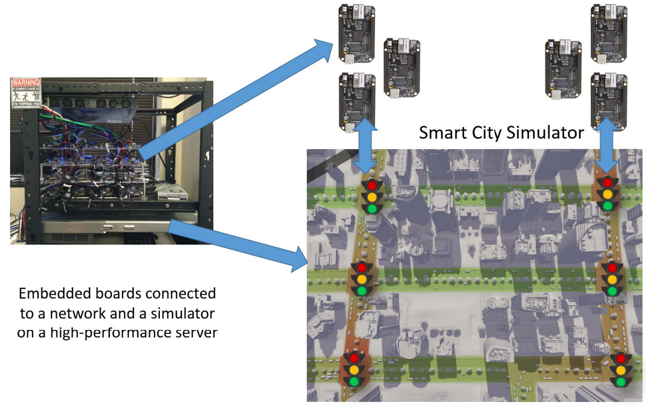

Fig. 3: Scenario showing the smart traffic lights and how they are controlled by embedded boards in the testbed with the rest of the system simulated by Cities: Skylines

cation network(s) and are connected to one or more real-time simulators. The simulation hosts provide the embedded computers the ability to sense and control the (simulated) physical systems in which they will be deployed. The configurable network emulator (through which the embedded computers communicate) is more accurate than a network simulator, as the hosts are using the same protocol stack and network interfaces as in the real world. For more information about this testbed please see [4]. An architectural diagram of the CPS testbed is shown in Figure 2.

Scenario: As an example of edge computing devices interacting with each other and the real world (as a CPS), consider a network of advanced traffic intersection controllers. These controllers have an array of local sensors connected to them by which they determine the state of the world around them. These sensors may be inductive loop sensors embedded in the road, proximity sensors for wireless devices, and cameras for image-based traffic monitoring. Furthermore, these devices do not simply control traffic lights, but also provide feedback and cooperative control to the vehicles on the roadways. Finally, these edge controllers communicate with each other to form a cooperative control group both in their local area and as part of a larger control hierarchy.

To enable the evaluation of such a complex system on the testbed, we had to find a suitable simulator that simulates with some fidelity the complex interactions. Since the controllers interact not only with each other but also the vehicles, pedestrians, and other edge computing nodes within proximity, the simulator must allow the configuration of and interaction with these simulated entities. Such a simulation allows the validation of edge computing devices and software applications for a new kind of Smart City.

Cities:Skylines: Cities:Skylines [1] provides the mechanism for simulation of the people and objects within an entire city, handling many hundreds of thousands of civilians simultaneously as they travel throughout the city by walking, driving, or taking some form of public transportation (or a combination of all three). Because Cities:Skylines was developed with a rich modding API, developers are free to create additions or replacements for the entities and the behaviors in the simulation. Specifically, for this demo scenario, we develop a HIL simulation for edge-computing-based, intelligent traffic lights that sense and communicate with the vehicles, pedestrians, and other local traffic lights to provide more coordinated and cooperative control of the intersections. In this scenario, the smart traffic light control algorithms execute on the embedded hardware of the testbed, while the physics and behavior of the vehicles and people of the city are simulated within Cities: Skylines. The scenario makes use of our custom network interface mod for Cities: Skylines which provides the API over the network; this network API enables the controller code on the testbed to query and configure the state of the traffic lights and query the information about the vehicles and the pedestrians (see figure 3).

Acknowledgment: The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000666 and in part by a grant from Siemens, CT. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or Siemens, CT.

REFERENCES

[1] Cities skyline. http://www.citiesskylines.com.
[2] Riaps. https://riaps.isis.vanderbilt.edu/redmine/projects/riaps.
[3] K. Benson, C. Fracchia, G. Wang, Q. Zhu, S. Almomen, J. Cohn, L. DâĂŹarcy, D. Hoffman, M. Makai, J. Stamatakis, et al. Scale: Safe community awareness and alerting leveraging the internet of things. *IEEE Communications Magazine*, 53(12):27–34, 2015.
[4] P. Kumar, W. Emfinger, and G. Karsai. Testbed to simulate and analyze resilient cyber-physical systems. In *Rapid System Prototyping, 2015. RSP '15.*, October 2015.
[5] L. M. Vaquero and L. Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *SIGCOMM Comput. Commun. Rev.*, 44(5):27–32, Oct. 2014.
[6] D. Willis, A. Dasgupta, and S. Banerjee. ParaDrop: A Multi-tenant Platform to Dynamically Install Third Party Services on Wireless Gateways. In *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*, pages 43–48. ACM, 2014.