Demo Abstract: DDFlow Visualized Declarative Programming for Heterogeneous IoT Networks on Heliot Testbed Platform

Joseph Noor, Sandeep Singh Sandha, Luis Garcia, Mani Srivastava

University of California, Los Angeles

Los Angeles, California

{jnoor,ssandha,garcialuis,mbs}@ucla.edu

ABSTRACT

Programming distributed applications in the IoT-edge environment is a cumbersome challenge. Developers are expected to seamlessly handle issues in dynamic reconfiguration, routing, state management, fault tolerance, and heterogeneous device capabilities. This demo presents DDFlow, a macroprogramming abstraction and accompanying runtime that offers appropriate distributed system tooling to properly isolate application semantics from arbitrary deployment environments. Using DDFlow leads to portable, visualizable, and intuitive applications. The accompanying runtime enables dynamic adaptation to improve end-to-end latency while preserving application behavior despite device failures. We evaluate DDFlow on the Heliot platform, a hybrid emulation testbed for learning-enabled IoT systems. This demo complements the paper "DDFlow: Visualized Declarative Programming for Heterogeneous IoT Networks" that is to be presented at IoTDI 2019 [6].

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; • **Computer systems organization** → **Distributed architectures**; **Sensors and actuators**; **Dependable and fault-tolerant systems and networks**; • **Software and its engineering** → **Software development methods**;

KEYWORDS

Macroprogramming, Declarative programming, Distributed systems, IoT networks, Visualized programming, Adaptation, Dynamic reconfiguration, Fault tolerance

ACM Reference Format:

Joseph Noor, Sandeep Singh Sandha, Luis Garcia, Mani Srivastava. 2019. Demo Abstract: DDFlow Visualized Declarative Programming for Heterogeneous IoT Networks on Heliot Testbed Platform. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '19)*, April 15–18, 2019, Montreal, QC, Canada. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3302505.3312598>

1 INTRODUCTION

The booming expansion of the Internet-of-Things has led to a world where advanced sensors, actuators, and specialized hardware can

interact with the environment in unprecedented ways, enabling emerging applications ranging from agriculture to city planning and military surveillance [1, 9]. A challenge arises in how to effectively specify and manage coordinated activity across a wide variety of heterogeneous hardware. For complex solutions derived from open-source tooling, developers are expected to build their own distributed systems. This often leads to a lack of portability, programmability, and efficient system management.

This demo introduces DDFlow (**Declarative Dataflow**), a macroprogramming abstraction and accompanying runtime that provides an efficient means to program high-quality distributed applications that span a diverse and dynamic IoT network. Application specification is accomplished through a declarative user interface implemented as an extension of the Node-RED IoT system [3]. Developers visually specify what the application should accomplish without explicit regards to the how. The accompanying DDFlow system runtime dynamically deploys applications to the available network by intelligently mapping tasks to available devices in order to minimize end-to-end latency. In the case of significant network changes or device failure, DDFlow will reconfigure to preserve application semantics by re-mapping the computation onto the network and/or switching to alternative networking protocols. In this demo, we realize an IoT scenario generated by DDFlow on the Heliot [8] platform.

2 DDFLOW IMPLEMENTATION

To support the DDFlow abstraction, we have developed a runtime implementation based on Node-RED [3], an open source framework that provides a graphical user interface for programming IoT devices. While Node-RED provides a single-device programming environment, we have built a distributed system atop the basic graphical interface, provided support for declarative DDFlow applications, and created system tooling for dynamic deployment and reconfiguration of applications spanning heterogeneous IoT networks. Please refer to [6] for a more complete description of the programming abstraction.

The system is comprised of two main architectural components, a *Coordinator* and *Device Manager*. The *Coordinator* provides a means for inter-device and application coordination, whereas the *Device Manager* provides a means for intra-device coordination, including starting and stopping relevant application tasks on a particular device.

The end-user interacts with a web interface hosted by the DDFlow *Coordinator*. The left block of Figure 1 is a screenshot of this interface presenting the specification of an example application. Each node in the DDFlow graph corresponds to at least one instance of a task that must run on a device in the network. Upon deployment,

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IoTDI '19, April 15–18, 2019, Montreal, QC, Canada

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6283-2/19/04.

<https://doi.org/10.1145/3302505.3312598>

Figure 1: Screenshot of the DDFlow user interface presenting an example application (see [6] for more information). DDFlow will send a JSON file that contains the IoT scenario to be emulated on the Heliot framework.

the system runtime elastically scales the application to the available network resources and monitors for significant environment changes. Upon detection of a major network change, the *Coordinator* recomputes a mapping of tasks to devices, optimizing for end-to-end application latency. In the case of significant latency reduction, the *Coordinator* adjusts a given application deployment.

Integration with Heliot. Figure 1 depicts how DDFlow interacts with Heliot. DDFlow provides a JSON file containing the IoT scenario to be emulated by the Heliot framework. Heliot can demonstrate the benefits of DDFlow’s abstraction for IoT macroprograms by adapting to dynamically changing environments.

3 DEMONSTRATION DESCRIPTION

To showcase the benefits of a declarative programming interface with an adaptive runtime, we developed a simulation testbed pictured in Figure 1. We utilize the Heliot framework [8] to emulate different IoT scenarios using both real IoT devices as well as simulated devices in the Airsim [7] environment simulator, all of which are connected via the Mininet [2] network emulator. Heliot allows us to emulate diverse network characteristics between real and virtual sensors/actuators. This provides a device-agnostic abstraction for the users of DDFlow.

The demo will consist of three visualization components: an interface allowing users to use DDFlow’s front-end to deploy macroprograms on the testbed, a display visualizing the simulated environments, and a display visualizing the impact of static deployment versus DDFlow’s dynamic placement algorithm. The motivating application example scenario consists of the following components:

- (1) A camera used to identify the target
- (2) Servers with heterogeneous compute capabilities (with or without GPU), and edge devices with local accelerators that can provide image classification
- (3) A client device to raise an alert upon target identification
- (4) A virtual drone that follows the target after identification
- (5) Emulated dynamic network characteristics between servers and the associated IoT devices.

Our demo will include a subset of the devices/sensors supported by Heliot (e.g., NVIDIA Jetson TX2 [4], Google Vision Kit [5], Raspberry Pi 3, smartphones, virtual Airsim sensors, Linux and Windows machines).

ACKNOWLEDGMENTS

Research reported in this paper was sponsored in part by the Army Research Laboratory (ARL) under Cooperative Agreement W911NF-17-2-0196, and by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, DARPA, SRC, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

- [1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2347–2376.
- [2] Ramon R Fontes, Samira Afzal, Samuel HB Brito, Mateus AS Santos, and Christian Esteve Rothenberg. 2015. Mininet-WiFi: Emulating software-defined wireless networks. In *Network and Service Management (CNSM), 2015 11th International Conference on*. IEEE, 384–389.
- [3] JS Foundation. 2018. Node-RED. <https://nodered.org>
- [4] D Franklin. 2017. NVIDIA Jetson TX2 Delivers Twice the Intelligence to the Edge. *NVIDIA Accelerated Computing Parallel Forall* (2017).
- [5] Google. 2018. Google Vision Kit. <https://aiyprojects.withgoogle.com/vision/>
- [6] Joseph Noor, Hsiao-Yun Tseng, Luis Garcia, and Mani Srivastava. 2019. DDFlow: Visualized Declarative Programming for Heterogeneous IoT Networks. In *Internet-of-Things Design and Implementation (IoTDI), 2019 IEEE/ACM Fourth International Conference on*. IEEE.
- [7] Shital Shah, Debadepta Dey, Chris Lovett, and Ashish Kapoor. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*. Springer, 621–635.
- [8] Sandeep Singh Sandha. 2018. nesl/Heliot. <https://github.com/nesl/Heliot>
- [9] Niranjan Suri, Mauro Tortonese, James Michaelis, Peter Budulas, Giacomo Benincasa, Stephen Russell, Cesare Stefanelli, and Robert Winkler. 2016. Analyzing the applicability of internet of things to the battlefield environment. In *Military Communications and Information Systems (ICMCIS), 2016 International Conference on*. IEEE, 1–8.