Poster: Dronemap - A Cloud-based Architecture for the Internet-of-Drones

Basit Qureshi
Prince Sultan University
Saudi Arabia
qureshi@psu.edu.sa

Anis Koubaa
Prince Sultan University, Saudi Arabia
CISTER/INESC-TEC,ISEP-IPP, Portugal
akoubaa@coins-lab.org

Mohamed-Foued Sriti
Al-Imam Mohamed bin Saud Islamic University
Saudi Arabia
mfsriti@ccis.imamu.edu.sa

Yasir Javed
Prince Sultan University
Saudi Arabia
yjaved@psu.edu.sa

Maram Alajlan
King Saud University
Saudi Arabia
maram.ajlan@coins-lab.org

Abstract

Unmanned Aerial Vehicles (UAVs) are opening new horizon as a major Internet-of-Things (IoT) player. UAVs are being used for several applications in surveillance, disaster management, search and rescue, environment monitoring, etc. Most of these solutions are limited to the point-to-point communication pattern, and are not suitable for applications in distributed multi-UAV scenarios. In addition, low-cost UAVs have limited processing and storage capabilities and cannot cope with massive computations requirements of certain applications. This paper proposes Dronemap, a cloud-based architecture for Internet-of-Drones (IoD), which integrates UAVs with the cloud to (1) virtualize access to UAVs, (2) offload heavy computations from the UAVs to the cloud.

Keywords

Internet-of-Drones, Cloud Computing, UAVs, Computation Offloading

1 Introduction

In recent years, there has been an increasing interest in the Internet-of-Things (IoT). We explore and focus on the use of Unmanned Aerial Vehicles (UAVs) to execute surveillance and security-related missions through a cloud computing infrastructure. This work garners motivation from limitations of typical low-cost UAVs having strict processing and storage capabilities constraints. Low-cost UAVs are unable to cope efficiently with the requirements of the surveillance and security applications, encompassing real-time data and reliability constraints. For example, the AR Parrot drone 2 is equipped with an embedded computer with a 1 GHz 32 bit processor and 1 Gbit DDR2 Random Access Memory (RAM) at 200 MHz. The presence of powerful on-board computers on every single UAV is cost-prohibitive and needless owing to the many limitations in the inherent design. Consequently, we aim at using the cloud as a remote brain for the UAVs by providing computation and storage services remotely [1]. The UAV simply acts as a mobile sensor node and/or sink that collects data from points of interest and transfer it to the cloud, which, in turn, stores, processes and provides interpretation for the collected raw data. In addition, the services provided by the cloud allow clients to connect, coordinate missions as well as initiate commands to the UAVs for future missions.

We propose Dronemap, a cloud based integrated architecture for mission control. Our approach leverages the use of cloud computing paradigm to address all the aforementioned requirements. There are two main benefits (1) Virtualization: the cloud infrastructure helps virtualizing UAV resources through abstract interfaces. It provides a mapping of the physical UAVs to virtual UAVs, so that end-users interact with virtual UAVs instead of the physical UAVs. This emphasizes the concept of the Internet-of-Drones (IoT) as a specific case of the IoT where the UAVs represent the things connected through the Internet through abstract interfaces. (2) Computation of- floading: the cloud plays the role of a remote brain for the UAVs by providing storage and computation services[1]. This approach overcomes the computing and storage resources limitations of the UAVs, as intensive computation is not performed on-board, but rather offloaded to the cloud. As such, the UAV will typically act as a mobile sensor and actuator decoupled from heavy computations. In what follows, we present the Dronemap architecture.

2 Dronemap Cloud Services Architecture

The overall objective of the UAV-cloud integration is to virtualize the access of UAVs through the cloud, and to offload computation from the UAVs to the cloud. Consider the following illustrative scenario: a team of multiple autonomous UAVs deployed in an outdoor environment in their depot waiting for the execution of certain missions. A user behind the cloud defines a mission (e.g. visiting a set of waypoints) and requests its execution. The user may either select one or more virtual UAVs from the list of available UAVs registered in the cloud, or may send his request to the

International Conference on Embedded Wireless
Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria

ISBN: 978-0-9949886-0-7

cloud to auto-select one or more UAVs to execute the mission. Each virtual UAV is mapped to a physical UAV by the cloud using a service-oriented approach, typically implementing SOAP or REST Web services. Once the mission request is received, the selected UAVs execute the mission and report in real-time data of interest to the cloud layer, which in turn will store, process and forward synthesized results to the user. Based on this scenario, we retain several challenges that can be addressed suing the dronemap architecture. Figure 1 presents the architecture of Dronemap addressing the above functional and non-functional requirements.

The UAV Layer: The UAV represents a set of resources exposed as services to the end-user. The UAV has several layers of abstractions. On top of the hardware, we opt for the use of ROS as a middleware that provides a first layer of abstraction that hides hardware resources (i.e. sensors and actuators). This allows software developers to focus more on the high-level development without having to deal with hardware issues. On top of ROS, two types of network interfaces are defined to expose ROS functionality to the outside: (1) *Web services interfaces:* allowing clients to access the UAV resources through Web services methods invocations, (2) *Cloud interfaces:* these are network sockets interfaces with JSON serialized messages to ensure platformindependent message exchange. Server applications are built on top of both network interfaces to address clients queries and execute their missions. The UAV communication through a wireless channel with the cloud.

Cloud Services Layer: Three sets of components are defined:(1) *Storage components:* This set of components provides storage for streams of data originated from UAVs and captured by this layer. Each UAVs environment variables, localization parameters, mission information, and transmitted data streams including sensor data and images with timestamps are stored in the cloud in a distributed file system (i.e. HDFS, NoSQL database such as HBase). This helps to perform largescale batch processing on stored data using tools like Hadoop Map/Reduce. Batch processing can be used to look for particular events into the log file, for example, how many intruders detected in unauthorized area over a certain period of time. In addition, real-time processing of incoming data streams is performed for detecting possible critical events of threats that require immediate action. (2) *Computation components:* Various computation intensive algorithms are deployed in the cloud. Image processing libraries process stored data available in HBase to detect possible event. In addition, Map/Reduce jobs running on the Yarn cluster allow applications to run in parallel reducing the processing time, therefore improving performance. Additionally Data Analytics algorithms can be executed on the stored set of large scale data. (3) *Interface components:* We defined three sets of interfaces as part of this component. (a.) *Network interfaces* implement network sockets interface on the server side. These provide listening to JSON serialized messages sent from UAVs. (b.) *The web services* allow clients to tweak UAV environment variables and mission control parameters. These interact with web-services interface in the UAV layer. (c.) *Data Analytics as a service* allows users to analyze stored data, and produce results as a service to the

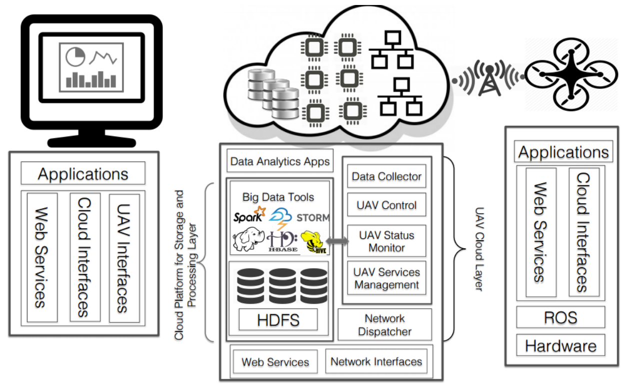

Figure 1. DroneMap System Architecture: Abstraction Layers

client.

Client Layer: The client layer runs *dronemap* client side application which provides interface to the cloud services layer as well as the UAV layer. Users have access to registering multiple UAVs, defining and modifying mission parameters and decision making based on data analysis provided by the cloud. The application allows users to monitor and control the UAVs and their missions remotely. Front-end interfaces provide the functionalities to the user to connect/disconnect, use available physical UAVs and their services, configure and control a mission and monitors the parameters of the UAV.

3 Discussion and Ongoing Work

In this paper, we presented an integrated cloud-based architecture for future Internet-of-Drones[2]. Various components of the *dronemap* project have been implemented. A cloud cluster with six nodes running Hadoop 2.6 on Ubuntu 14.4 LTE operating system was deployed. As part of the cloud services layer various libraries were written in Java and Python for connectivity to UAVs, receiving data and logging into the HBase table. Furthermore, a Robot-webtools library was developed for ROS allowing connectivity with AR.Parrot 2.0, Bebop Parrot 3 as well as Earle-copter drones. Transmission to the cloud cluster was successfully tested over the local area network. We are currently working towards the deployment of the UAVs over the Internet through the *dronemap* cloud infrastructure, and we are developing protocol for cloud-UAV interaction, namely, auto-discovery of drone, cloud registration, Web services access to drones, mission planner, and other services.

4 Acknowledgments

This work is supported by a King AbdulAziz City for Science and Technology (KACST) under grant number 35-157, and supported by Prince Sultan University.

5 References

[1] E. Guizzo. *Robots with their heads in the clouds*. Spectrum, IEEE,48(3):1618, March 2011.

[2] L. Mottola *Real-world drone sensor networks: A multi-disciplinary challenge*. In Proceedings of the 6th ACM Workshop on Real World Wireless Sensor Networks, RealWSN 15, pages 11, New York, NY, USA, 2015. ACM.