Check for updates

Poster Abstract: Scalable Distributed Microservices for Autonomous UAV Swarms

Kevyn Angueira Irizarry, Zichen Zhang, Christopher Stewart
Ohio State University

Jayson Boubin
Binghamton University

ABSTRACT

Unmanned aerial vehicles (UAV) are revolutionizing critical industries. Their inexpensive and accessible nature makes them useful for a number of broad applications including agriculture, infrastructure inspection, and more. In response to this popularity, UAV manufacturers, hobbyists, and researchers have developed myriad software packages for UAV control to simplify and automate UAV flight. Recent advances have also led to autonomous UAV that complete complex missions without human pilots and swarms of UAV that work together to solve tasks. Recently, researchers have used autonomy and swarms to allow UAV to cover wide areas quickly and intelligently. Few software packages explicitly support either autonomy and swarming for UAV, and none to our knowledge combine these features. We present early work on SoftwarePilot 2.0, a UAV software package that supports swarms of autonomous UAV. SoftwarePilot 2.0 improves on prior work to expand microservice model designs which are easier to manage using cloud-native technologies. SoftwarePilot 2.0's edge-efficient design allows UAV swarms to easily scale across the edge and cloud, and supports cutting edge autonomy techniques.

CCS CONCEPTS

• **Applied computing** → **Agriculture**; • **Computer systems organization** → **Distributed architectures**.

KEYWORDS

unmanned aerieal vehicles, edge computing, swarm, agriculture

ACM Reference Format:

Kevyn Angueira Irizarry, Zichen Zhang, Christopher Stewart and Jayson Boubin. 2022. Poster Abstract: Scalable Distributed Microservices for Autonomous UAV Swarms. In *23rd International Middleware Conference Demos and Posters (Middleware '22 Demos and Posters), November 7–11, 2022, Quebec, QC, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3565386.3565484

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are inexpensive and maneuverable IoT devices which are quickly changing key industries. UAV can sense over wide areas quickly in 3D space, access areas too dangerous for humans, and can act in groups as swarms to distribute tasks and learn from one another. Recent work has shown that UAV contribute broadly to agriculture, construction, infrastructure inspection, search and rescue, and more [9, 10].

While UAV are conventionally piloted by humans, improvements in onboard hardware and control software have led to automated and autonomous UAV flights. Open source and manufacturer-provided software development kits (SDKs)[4, 8], support libraries [7], and research middlewares [2, 6, 11, 12] provide features for UAV flight control via waypoint missions, data capture, computer vision, and even complex decision-making. Autonomous UAV flight, where UAV make complex decisions in-mission, has been shown to speed up some mission types and allow for new more complex UAV missions [3].

Despite their benefits, UAV suffer from critical bottlenecks. UAVs have short flight times due to small batteries, and limited onboard compute capacity. To cover wide areas, distribute intelligence, and elongate missions, UAV are often flown in swarms. Swarms of UAV can be dispatched across wide areas, learn from each other's observations, and work together to solve complex problems. Swarms of autonomous UAV can accomplish complex tasks quickly and benefit from each other's observations.

Few software packages provide even baseline UAV swarm capabilities [5]. Similarly, few packages and SDKs provide capabilities for autonomy [2, 11]. Some provide simple computer vision routines and automated flight, but none provide support for custom autonomy policies, computer vision, inter-UAV communication, or swarm control. Even complex research middlewares for UAV rarely provide native swarm support or combine it with autonomy. This is because autonomous swarms are hard to manage and scale. Single autonomous UAV already necessitate complex edge hardware and resource management practices [3]. Scaling up from a single UAV to a swarm includes not only considering resource impacts of additional swarm members, but how those impacts compound as members share information and learn.

In this paper, we present our adaptations to an autonomous UAV middleware, SoftwarePilot, which provides mechanisms for creating autonomous UAV. We modified SoftwarePilot to work with cloud-native and edge-appropriate scalable deployment technologies. SoftwarePilot 2.0 leverages these capabilities to deploy, distribute, and manage autonomous UAV swarms across edge clusters.

2 DESIGN

SoftwarePilot [2] is a UAV middleware that allows users to implement autonomous missions. As shown in Figure 1, SoftwarePilot decouples mission code into loadable microservices called routines and drivers linked by the SoftwarePilot API. Drivers are application specific APIs that control UAV from different manufacturers, supply pathfinding and AI algorithms, and manage data. Routines are user-code that access drivers via the SoftwarePilot COaP API.

Middleware '22 Demos and Posters, November 7–11, 2022, Quebec, QC, Canada

ACM ISBN 978-1-4503-9931-9/22/11.
https://doi.org/10.1145/3565386.3565484

This design already includes scalable elements. By decoupling APIs from user code, users can load and unload drivers dynamically based on their needs. SoftwarePilot's original design did not, however, consider a number of important management concerns relating to UAV swarms. First, SoftwarePilot 1.0 UAVs are controlled by a single central virtual machine on which all microservices run, making it portable but difficult to distribute. Second, while SoftwarePilot was made to build autonomous UAV, it was not made to operate swarms. It includes no deployment mechanisms for swarms across edge clusters, network overlay features, or distributed systems management software. Third, it does not include any services for leveraging swarm intelligence.

We addressed these three shortfalls in SoftwarePilot 2.0. First, we converted all SoftwarePilot microservices from independent applications to Docker containers. This change is significant in that it allows for increased portability without virtual machine overhead, it decentralizes microservices from inside the virtual machines where they previously ran, and allows SoftwarePilot to benefit from existing container deployment technologies. SoftwarePilot uses Kubernetes to deploy its containers across clusters of edge devices. Kubernetes features allow SoftwarePilot containers to communicate via overlay networks, set resource limitations, deploy to specific cluster nodes, and manage the experiment lifecycle in ways that our prior virtualization technique did not.

SoftwarePilot has also added support for UAV swarms and multi-agent reinforcement learning (MARL). SoftwarePilot 1.0 had no inherent way to instantiate multiple UAV and allow them to communicate. Using the new SoftwarePilot Swarm Intelligence API, users can instantiate and control multiple UAV, link them to microservices, and control them via centralized or distributed programming logic. This new API and Python package can leverage the kubernetes overlay network to communicate with UAV control and autonomy microservices across the cluster.

Additional swarm intelligence APIs can use this inter-UAV communication to implement global multi-agent reinforcement learning policies. SoftwarePilot 1.0 implemented autonomy policies on a drone by drone basis. SoftwarePilot 2.0 now allows users to implement MARL policies on top of single-UAV autonomy policies. MARL policies can track reward and mission progress at a global level. They combine observations from multiple agents to make better decisions and retrain models over time.

3 EARLY EXPERIENCES

SoftwarePilot 2.0 has been used to implement UAV swarm applications in agriculture. In Autumn 2021, we flew over 150 UAV Swarm missions using SoftwarePilot 2.0's MARL and swarm features over crop fields in Ohio [1]. We used a swarm of autonomous UAVs to sample a crop field to assess soybean leaf defoliation, an important crop health indicator. We used SoftwarePilot drivers for the DJI Mavic UAV combined with custom drivers for defoliation detection using DefoNet [13]. To build an intelligent UAV swarm, we developed SoftwarePilot's swarm intelligence APIs. We implemented APIs for multi-agent Q-learning and online model updating which were containerized and distributed across our edge hardware using Docker and Kubernetes.

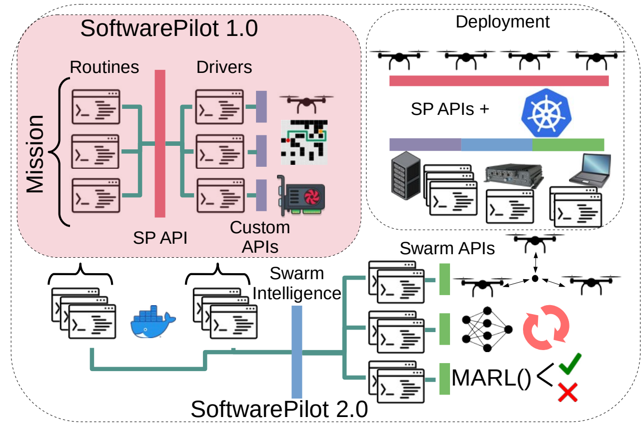

Fig. 1: SoftwarePilot 2.0 adds new swarm intelligence APIs and deployment management.

SoftwarePilot 2.0's new swarm mechanisms are efficient. For our deployment configuration, we found that SoftwarePilot 2.0 used 2X less energy than a swarm created using SoftwarePilot 1.0. This was due entirely to the use of our Kubernetes management platform which automatically duty-cycles cluster resources during resource troughs. We also found that the addition of MARL as sped up missions by up to 2.1X via better decision-making. In the near future, we plan to release SoftwarePilot 2.0 as an open source project to help facilitate the building and deployment of fully autonomous UAV swarms.

Acknowledgments: This work was funded by NSF Grant OAC-2112606 and the Ohio Soybean Council.

REFERENCES

[1] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart. Marble: Multi-agent reinforcement learning at the edge for digital agriculture. In *Proceedings of the 7th ACM/IEEE Symposium on Edge Computing*, 2022.
[2] J. Boubin, C. Stewart, S. Zhang, N. T. Babu, and Z. Zhang. Softwarepilot. http://github.com/boubinjg/softwarepilot, 2019.
[3] J. G. Boubin, N. T. Babu, C. Stewart, J. Chumley, and S. Zhang. Managing edge resources for fully autonomous aerial systems. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 74–87. ACM, 2019.
[4] DJI. Prerequisites-dji mobile sdk documentation. https://developer.dji.com/, 2018.
[5] EpiSci. Swarmsense: Intelligent uav swarms. https://www.episci.com/product/swarmsense/.
[6] H. Genc, Y. Zu, T. Chin, M. Halpern, and V. J. Reddi. Flying iot: Toward low-power vision in the sky. *IEEE Micro*, 37(6):40–51, November 2017.
[7] L. Meier, J. Camacho, B. Godbolt, J. Goppert, L. Heng, M. Lizarraga, et al. Mavlink: Micro air vehicle communication protocol. *Online]. Tillgänglig: http://qgroundcontrol. org/mavlink/start.[Hämtad 2014-05-22]*, 2013.
[8] Parrot. Parrot olympe sdk. https://developer.parrot.com/docs/olympe/index.html.
[9] C. Qu, J. Boubin, D. Gafurov, J. Zhou, N. Aloysius, h. Nguyen, and P. Calyam. Uav swarms in smart agriculture: Experiences and opportunities. In *2022 IEEE International Conference on eScience*. IEEE, 2022.
[10] L. Ruetten, P. A. Regis, D. Feil-Seifer, and S. Sengupta. Area-optimized uav swarm network for search and rescue operations. In *2020 10th annual computing and communication workshop and conference (CCWC)*, pages 0613–0618. IEEE, 2020.
[11] J. L. Sanchez-Lopez, R. A. S. Fernández, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy. Aerostack: An architecture and open-source software framework for aerial robotics. In *International Conference on Unmanned Aircraft Systems*, 2016.
[12] A. Van't Hof and J. Nieh. Androne: Virtual drone computing in the cloud. In *Proceedings of the Fourteenth EuroSys Conference 2019*, pages 1–16, 2019.
[13] Z. Zhang, S. Khanal, A. Raudenbush, K. Tilmon, and C. Stewart. Assessing the efficacy of machine learning techniques to characterize soybean defoliation from unmanned aerial vehicles. *Computer and Electronics in Agriculture*, 2022.